AMSTRAD MAGAZINE

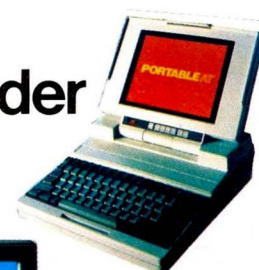
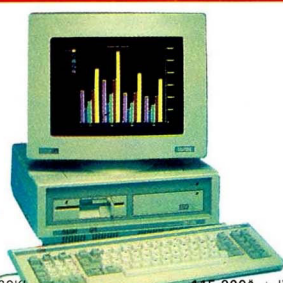
REVISTA DOS UTILIZADORES AMSTRAD

REVISTA MENSAL N.º 15/16 ANO 2 NOVEMBRO/DEZEMBRO 1989 350$00

TESTE

DECO RECOMENDA

AMSTRAD PC 1640 É A «ESCOLHA ACERTADA» NA COMPRA DE UM COMPUTADOR PESSOAL

Alfa sistemas
Informática e Burótica, Lda.

1512 c/ 2 drives de 360Kb .... 115.900$ + IVA
1512 c/ 1 drive de 360Kb + disco de 20Mb ...... 180.500$ + IVA
Monitor Monocromático .... 38.950$ + IVA
Monitor Policromático .... 75.050$ + IVA

1640 c/ 2 drives de 360Kb .... 133.000$ + IVA
1640 c/ 1 drive de 360Kb + disco de 20Mb ...... 213.750$ + IVA
Monitor Monocromático .... 41.800$ + IVA
Monitor Policromático EGA .... 114.000$ + IVA

Portátil c/ 2 drives de 720Kb + modem .... 165.000$ + IVA

GAMA AMSTRAD 2000

2086 c/ 1 drive de 720kb .... 175.000$ + IVA
2086 c/ 2 drives de 720 Kb .... 190.000$ + IVA
2086 c/ 1 drive de 720Kb + disco de 30Mb ...... 295.000$ + IVA

2286 c/ 2 drives de 1,44Mb .... 295.000$ + IVA
2286 c/ 1 drive de 1,44Mb + disco de 40Mb .... 430.000$ + IVA
2386 c/ 1 drive de 1,44Mb + disco de 65Mb .... 795.000$ + IVA
Monitor Monocromático VGA de 12'' .... 35.000$ + IVA
Monitor Policromático VGA de 14'' .... 80.000$ + IVA
Monitor Monocromático VGA HR de 12'' ... 110.000$ + IVA
Monitor Policromático VGA HR de 14'' .... 145.000$ + IVA

EURO PC
- Ecran Monocromático ....
- Ecran a Cores ....
- FDD (Drive adicional) 360 KB ....
- FDD (Drive adicional) 720 KB ....
- Disco Duro 20 MB ....

PORTÁTIL AT ....

Personal Fax SPF 200 ..... 200.000$ + IVA
Personal Fax SPF 100 ..... 187.500$ + IVA

TOWER-AT

Ecran Monocromático
- AT-201 (1D) ....
- AT-202 (2D) ....
- AT-220 (1D+Disc 20 MB) ....
- AT-260 (1D+Disc 60 MB) ....

C/ Ec. a cores 14" CGA mais ....
C/ Ec. a cores 14" EGA mais ....
C/ Ec a cores 14" Multisyncrono mais

☎ TELEFONE-NOS

ETC.

DISQUETTES E OUTROS CONSUMÍVEIS

MEMORY — DS / DD 5.25'' 48 TPI .... 85$5 + IVA
SELECT — DS / DD 3.5'' 135 TPI .... 2.821$ + IVA

SOFTWARE

OPEN ACESS II PLUS (PROGRAMA)
CONTABILIDADE, FACTURAÇÃO, STOCKS, ETC.

MODEMS

- **PC CARD MODEM** (2400 BAUD) ...... **45.505$** + IVA
- **MODEM** (2400 BAUD) ...... **56.905$** + IVA

IMPRESSORAS

AMSTRAD

DMP3160 IMPRESS. 160 CPS-80/132 COL. NQL .... **50.255$** + IVA

DMP4000 IMPRESS. 200 CPS-132/230 COL. NLQ .... **85.405$** + IVA

LQ3500DI IMPRESS. 160 CPS-80/132 24 AGULHAS .... **83.505$** + IVA

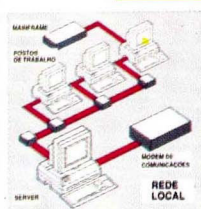
LQ5000DI IMPRESS. 288 CPS-132/230 24 AGULHAS .. **132.905$** + IVA

SCHNEIDER

43134 PRINTER 180 9 AGULHAS .... **42.750$** + IVA

43156 PRINTER 264 24 AGULHAS ...... **159.125$** + IVA

43180 LASER PRINTER **275.000$** + IVA

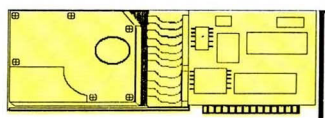
REDES LOCAIS

REDE AMSTRAD (KIT P/ SERVER + 3 POSTOS .... **111.050$** + IVA

REDE D/ LINK (KIT P/ SERVER + 3 POSTOS .... **129.000$** + IVA

REDE LANSMART ETHERNET (KIT P/ SERVER + 3 POSTOS) ... **212.000$** + IVA

PLACAS

NOVIDADE

UM DISCO PARA O SEU PC
HARDCARD DE 40 MBYTES

A 40 MILISEGUNDOS
SÓ **75.000$00** + IVA
E nós instalamos!

ENVIAMOS À COBRANÇA PARA TODO O PAÍS
PREÇOS ESPECIAIS PARA REVENDA

DATA-SHOW

SISTEMA KODAK

AUMENTE O ALCANCE DO PC PROJECTANDO-O NUM ECRAN GIGANTE

DESDE **150.000$** + IVA

Calçada do Carmo, n.º 18 - 1.º 1200 LISBOA Tel. 32 29 92 — RETOMA DE MATERIAL USADO

PROPRIEDADE:
PUBLINFOR, Publicações e Comércio de Artigos de Informática, S.A.
R. São Gonçalo, 967
4800 GUIMARÃES

REDACÇÃO, ASSINATURAS, PUBLICIDADE E "CLUBE AM":
Av. da Boavista, 2881—1.º
4100 PORTO
Telefs. 675395/673992
Telex 27250 P-Fax 678784

DIRECTOR:
Nunes Carneiro

COLABORADORES:
André Campos
António Cardoso
António Torres Martins
Carlos Guerreiro
João Paulo
João Pereira
Jorge Ramalheira
Margarida Santoalha
Maria de Lurdes Leite
Mário Leite
Paulo Pinheiro
Rui Mota

REVISÃO:
Fernando Silva

SECRETARIADO:
Carla Fonseca
Josefa Gonçalves

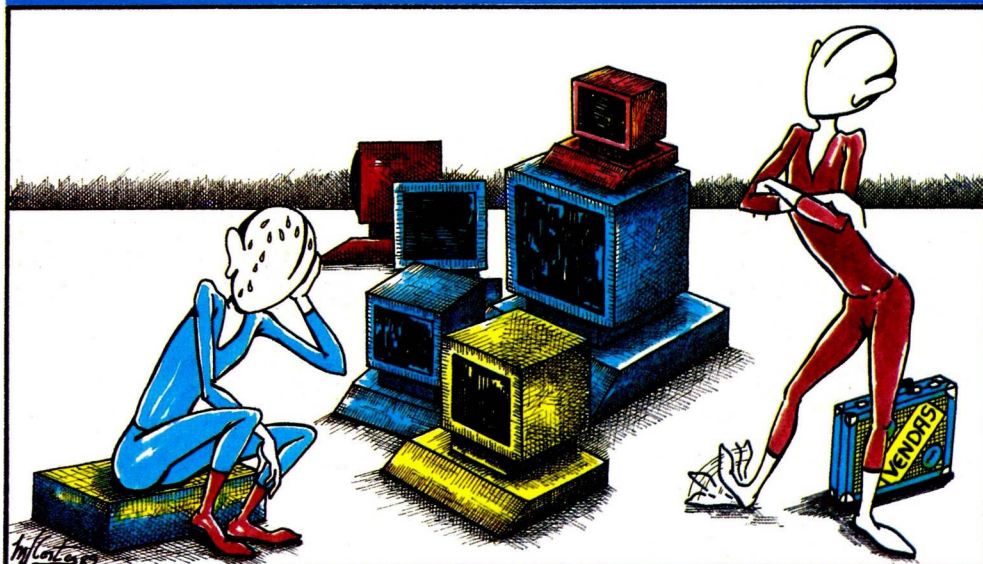
"CLUBE AM"
Luisa Martins

EXECUÇÃO GRÁFICA:
EDIÇÕES ASA—DIVISÃO GRÁFICA

DISTRIBUIÇÃO:
ELECTROLIBER

TIRAGEM: 11 500 exemplares

N.º PESSOA COLECTIVA: 502.009.870
N.º REGISTO D.G.C.S.: 112.959

A ESCOLHA DE UM COMPUTADOR

1. O número de pessoas interessadas em adquirir um computador pessoal é cada vez maior. E são também inúmeras as opções disponíveis no mercado nacional. Mas, no fundamental, continuamos a ser consumidores mal informados.

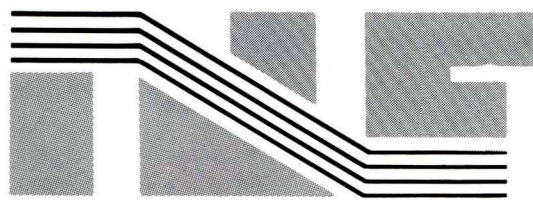
Muitos utilizadores (ou potenciais utilizadores) sentem-se, frequentemente, confusos e os meios de informação são escassos.

Neste contexto, assume particular importância a publicação pela DECO—Associação Portuguesa para a Defesa do Consumidor, de um teste sobre "computadores de uso pessoal". A DECO analisou e testou 15 modelos de computadores pessoais de diversas marcas, segundo diversos critérios: sistema operativo, monitor, velocidade, teclado, facilidade de utilização, etc..

Os resultados e a leitura deste teste são de inegável interesse para quem queira adquirir o seu computador.

2. Local privilegiado de contacto com o mercado informático português é a INFORPOR, que este ano se realiza em Lisboa. A maior feira de informática em Portugal é, talvez, o local ideal para todos os que desejam adquirir um computador. Aí estarão presentes as principais marcas com os seus inúmeros modelos. Óptima oportunidade para experimentar, testar e comparar. Para que a escolha do seu computador seja esclarecida e consciente.

Nunes Carneiro

SUMÁRIO

TOTOLOTO

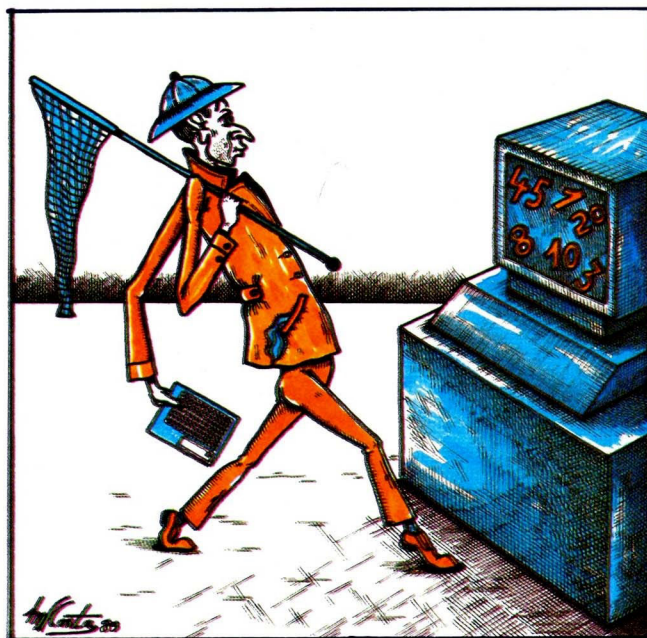
O COMPUTADOR AJUDA A GANHAR

O totoloto é uma verdadeira febre em Portugal. Todos os sábados, um pouco antes das 20 h, milhares (ou milhões?) de Portugueses sentam-se em frente do televisor e esperam os seis números, ou melhor, os seus seis números.

Consciente desta situação,uma empresa de Matosinhos (Fernandes M.ª Crista—Informática) criou um programa para ajudar os apostadores a acertar.

O programa, além de fazer o desdobramento automático de milhares de apostas, imprime ainda os boletins. Segundo Carlos Fernandes , o ''pai'' do ''Jackpot'', o programa é a solução informática para ter a certeza de acertar no totoloto.

Lançado em Janeiro deste ano, o sucesso do ''Jackpot'' tem sido assinalável.

Por 19 900$00, a empresa tem vendido um por dia.

Será que você também vai querer entrar nesta onda de certeza no mundo da sorte?

FINANTIAL MANAGER/FIXED ASSETS

PROMOSOFT apresentou ''packages'' de gestão financeira concebidos por países da CEE

A PROMOSOFT—Serviços Informáticos, Lda., levou a efeito um seminário para apresentação do FM (Finantial Manager) e FA (Fixed Assets), dois ''packages'' de gestão financeira, criados pela empresa italiana FORMULA, tendo em vista as regras legais e fiscais da Comunidade Económica Europeia.

O Finantial Manager (FM) e Fixed Assets (FA) apresentam-se com características fortemente inovadoras, sendo de realçar que o utilizador possui um controlo total do sistema, uma vez que a própia ''customização'' do produto é efectuada através de transacções ''on-line'' e sem compilações.

Sistema modular, cujos elementos podem correr em conjunto ou separadamente, possui componentes avançados de ''query'' e ''reporting''

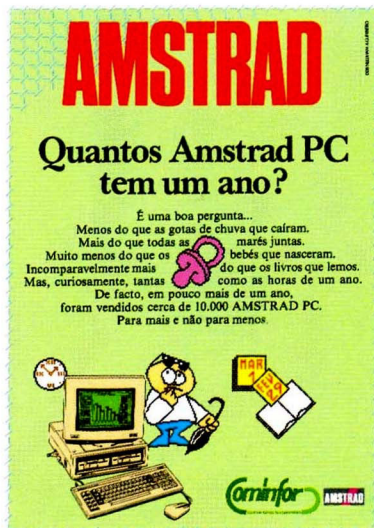
TELECOMUNICAÇÕES CTT ADQUIREM SOFTWERE DE GESTÃO DE BASE DE DADOS

As TELECOMUNICAÇÕES CTT assinaram, em 18 de Agosto último, um contrato global com a ORACLE CORPORATION, para fornecimento de Software de Gestão de Bases de Dados.

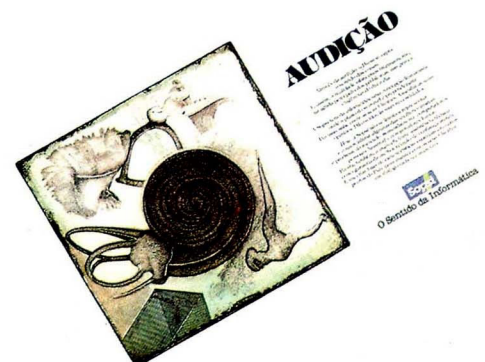
O Software fornecido pela ORACLE às TELECOMUNICAÇÕES CTT vai permitir a compatibilização de diferentes plataformas de hardware, nomeadamente ao nível da conectividade e da portabilidade total das aplicações.

Sediada na Califórnia, a ORACLE é o maior fornecedor mundial deste tipo de software. Está representada em Portugal pela Compta— Equipamentos e Serviços de Informática, S.A.

LIVRO DE OURO DA PUBLICIDADE

AMSTRAD E SOPSI EM DESTAQUE

A Amstrad e a SOPSI têm grande destaque na edição de 1989 do ''Livro de Ouro da Publicidade''. Com efeito, secção ''Material de Escritório'' desta publicação da revista ''Marketing e Publicidade'', estão presentes quase todos os anúncios da Amstrad e da SOPSI publicados na imprensa durante 1989.

De parabéns estão pois, a Cominfor e a SOPSI e, muito particularmente, a agência do grupo: a DDB Needhan & Guerreiro.

No ''Livro de Ouro'', a nível do mercado informático, só estão presentes apenas mais duas marcas: a IBM e a Rank Xerox.

NOVA FÁBRICA EM ESPANHA

ONDYNE APOSTA NO MERCADO IBÉRICO

A Ondyne, uma das mais empresas europeias de sistemas de alimentação ininterrupta, inaugurou recentemente uma nova fábrica em Colmenar Viejo, nos arredores de Madrid.

A Ondyne, prepara-se para exportar 82% da produção desta nova unidade para mais de 30 países.

CONQUISTAR O MERCADO EUROPEU E PENSAR NOS E.U.A.

A Ondyne possui já filiais em vários países europeus (Espanha, França, Inglaterra, Holanda, Alemanha Federal, Bélgica, Grécia, Portugal) e prevê a abertura de outras na Itália e nos países nórdicos. Em estudo está também a associação com um distribuidor nos Estados Unidos.

Os novos produtos exportados de Espanha foram investigados, desenvolvidos e produzidos no país vizinho. O seu objectivo é a conquista de uma posição relevante no mercado das gamas pequena e média.

São quatro as orientações estratégicas da Ondyne para os próximos anos: ''Implantar-se geograficamente nos países europeus onde ainda não haja uma filial da Ondyne''; ''entrar no mercado dos produtos de desenvolvimento avançados que estamos a preparar para as gamas méida/alta e alta''; ''incrementar fortemente as cotas de mercado onde já se comercializam os produtos Ondyne, graças à qualidade dos novos produtos que desenvolvemos''; e, finalmente ''a médio prazo, entrar no mercado dos Estados Unidos da América''.

A MAIOR FÁBRICA EUROPEIA

A fábrica agora inaugurada é a maior, a nível europeu a maior fábrica de sistemas de alimentação ininterrupa. Desta unidade sairão todos os produtos Ondyne para as filiais dos países europeus. E, no seu Departamento de Investigação e Desenvolvimento, desenvolvem-se já os novos produtos da gama Ondyne.

Em Colmenar Viejo trabalham 116 pessoas, assim repartidas por sectores: produção (72), laboratório (11), administrativos (2*), serviços de pós-venda (3), marketing e comercial (10).

Entre 1986 e 1989, a quantidade de equipamentos vendidos quintuplicou (Quadro 1).

Esta evolução, aparentemente surpreendente, tem uma explicação: ''as necessidades de dispor de um recurso opcional à energia eléctrica estão a aumentar e a diversificar-se''. Não só na área da informática como também nas aplicações médicas (aparelhos de análise médica, unidades ciruúrgicas, etc.), nas aplicações científicas (laboratórios de física, química, energia nuclear e aeronáutica, etc.), nos aeroportos, nos sistemas automáticos de produção, etc.

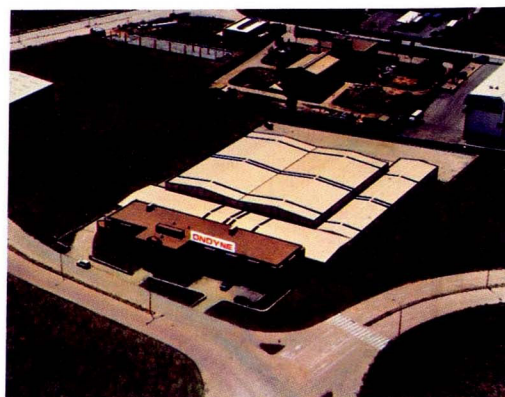
No campo da micro-informática, o índice de penetração dos sistemas de alimentação ininterrupta não pára de crescer. Isto é, a quantidade destes sistemas em relação ao número de micro-computadores cresceu, só em 1989, cerca de 8 por cento.

Coincidindo com a inauguração da nova unidade fabril a Ondyne apresentou a sua nova gama de produtos: a Multipower, a última geração de sistemas de alimentação ininterrupta.

A linha ''Power Lab H0'', constituída por 4 modelos, destina-se ''a todos os tipos de computador e permite eliminar totalmente o tempo de transferência da rede ao sistema e vice-versa''. Os ''Power Lab H0'' eliminam ''cortes, interrupções de corrente, oscilações de tensão e protegem os ficheiros de qualquer perda de informação ou danos no equipamento''. A corrente é filtrada e revelado quando não existem cortes na tensão de 220 volts.

O ''Power Center H0'' tem um tempo de comutação de 0 segundos. Situa-se entre o computador e o monitor. O ''Power Center H0'' comanda as conexões e desconexões entre o computador, monitor e periféricos (impressoras, plotters, etc.). Com a opção ''Alarme'' (Software de Gestão Power Centre), quando há o corte de corrente, o Power Center recebe um sinal que assinalara a anomalia e o estado das baterias. No caso de estas se esgotarem, cessa a aplicação em curso no computador, sendo actualizados automaticamente os ficheiros. A aplicação pode ser utilizada normalmente logo que o computador volte a ser alimentado pela rede.

EM PORTUGAL: MATIEX

A estrada da Ondyne em Portugal fez-se através da sua associação com a Mantiex, com sede em Gaia. A receptividade dos produtos Ondyne em Portugal tem sido ''muito boa'', segundo nos revelou Manuel Vasconcelos, Director Comercial da Matiex.

MEIA MARATONA DE NAZARÉ

Realizou-se no passado dia 12/11 a 15.ª edição da Meia-Maratona da Nazaré, que reuniu à partida mais de 3.500 atletas Nacionais e Estrangeiros. O tratamento informático desde as inscrições às classificações foi executado por computadores Amstrad. Mais uma vez a Cominfor fez deslocar o seu camião — Amstrad àquela típica vila piscatória, onde os milhares de espectadores colocados ao longo da marginal e indiferentes à chuva que caía com realizações e onde a organização sempre impecável e eficiente montou «Quartel General» para a Imprensa.
Que venha a edição de 90!

VI EXPOSIÇÃO INTERNACIONAL DE INFORMÁTICA E DE TECNOLOGIAS DA INFORMAÇÃO

INFORPOR 89

Após seis anos de realizações sucessivas, com início em 1984, no Porto, vai este ano ser pela primeira vez apresentada em Lisboa, nas instalações da FIL, entre 26 e 29 de Outubro, a INFORPOR 89, que de futuro passará, alternadamente, a realizar-se numa e noutra cidade.

Como anteriormente, trata-se da mais importante exposição-feira de equipamento de informática realizada em Portugal, vindo a verificar-se, por este motivo um interesse crescente, quer do lado dos expositores—este ano contar-se-à com a participação de cerca de 200—quer do próprio público e profissionais do ramo que cada vez mais procuram manter-se actualizados com os progressos do desenvolvimento das modernas tecnologias de informação.

A INFORPOR 89 volta a ser uma realização conjunta da Certame e da Associação Portuguesa de Informática, tendo este ano o apoio da Associação Industrial Portuguesa.

BANCO DE PORTUGAL

INSTALA NOVO SISTEMA DE CONTABILIDADE

O Banco de Portugal assinou um contrato com a Management Science America (MSA), através da sua representante Redware, para instalar um novo sistema de contabilidade.

O vice-presidente da MSA para a Europa, Martin Judd, esteve em Lisboa para assinar o contrato entre o Banco de Portugal e aquela empresa, uma das maiores companhias de software do mundo.

O novo sistema a ser instalado, no valor de 30 mil contos, distingue-se do anterior usado pelo Banco de Portugal pela grande flexibilidade e potência em informações de controle e gestão.

A Redware, lançada e comparticipada pela Reditus, dedica-se à prestação de serviços de informática no mercado de sistemas de médio e grande porte.

A Redware é representante exclusiva em Portugal de toda a gama de software de aplicação da MSA Internacional, a maior multinacional de software do mundo no campo dos mainframes (grandes computadores), com uma facturação anual superior a 250 milhões de dólares.

''A inovação do software fornecido pela MSA provém do facto dos clientes poderem comprar, por um preço relativamente baixo, programas que custaram fortunas a desenvolver, e no qual colaboraram grandes equipes de especialistas'' disse o vice-presidente da MSA.

Não seria rentável para qualquer empresa desenvolver ela própria este tipo de software, acrescentou Martin Judd.

Além disso, no caso da CEE, por exemplo, as necessidades são cada vez mais normalizadas, pelo que é possível desenvolver programas muito avançados capazes de desenvolver os problemas em todos os países membros, com pequenas modificações locais, adaptadas às necessidades de cada empresa ou cliente. A MSA fornece um serviço contínuo de acompanhamento dos sistemas implantados, com apoio técnico, documentação e ensino, no sentido de assegurar a eficácia do equipamento.

PERSONAL COMPUTER SHOW'89

NOVIDADES AMSTRAD

De 29 de Setembro a 1 de Outubro realizou-se, em Londres, a vigésima edição do Personal Computer Show. Em Earls Court apareceram mais de 400 expositores que mostravam os seus produtos desde, como é óbvio, os computadores pessoais até toda a espécie de acessórios e de software para eles disponíveis.

Com a habitual organização tendente a separar os produtos profissionais das máquinas e software de jogos, esta exposição continua a não deixar de parecer mais do que um grande bazar de brinquedos com um escritório, onde se podem encontrar alguns produtos de informática.

Do exposto, é fácil entender-se que o Show não é atractivo para os fabricantes apresentarem grandes novidades, guardando as primeiras mostras dos seus produtos para outras feitas, como a CEBIT ou a CICOB, a nível europeu.

Assim, só dois produtos despertaram a nossa curiosidade e, por acaso, dois portáteis: um Machintosh e o Móbil Computer da PSION.

O primeiro, por há muito se aguardar uma versão portátil do já célebre Mac.

O segundo, pelas suas dimensões. Realmente, um sistema capaz de trabalhar em MS-DOS (versão 3.2) com um peso inferior aos 2 Kg é digno de nota.

E a AMSTRAD?

També lá estava. Logo à entrada, com um imponente e majestoso stand para mostrar algumas, mas poucas novidades.

O PC 1386 que mais não é do que uma versão reduzida do modelo equivalente da série 2000, dado a versão base a partir de 1 Mb de memória RAM e não ter memória cache. No entanto, a expansibilidade do modelo leva-o às mesmas performances do 2386, ou seja, aos 16 Mb de memória RAM e aos 64 Kb de memória cache.

O PC 1286, também presente, é que não traz nada de novo, dado as suas características técnicas serem rigorosamente as mesmas do 2286. O porquê da nova referência deve-se exclusivamente ao facto desta(série (1286/1386) ter uma maior gama de unidades de disco disponíveis:

PC 1286	PC 1386
40 Mb	40 Mb
60 Mb	60 Mb
—	150 Mb

A SP 200 era uma das pseudo-novidades presentes no stand da Amstrad. No fundo, a SP 200 nada mais é do que a conhecida DMP 3160, mas na cor negra para não contrastar com os Sinclair PC 200, modelo para o qual é especialmente destinada.

Até aqui, nada é realmente novo e tudo se resume à troca de referências. O que chamou em especial atenção da AM foi a, já há tanto esperada, fax da AMSTRAD.

Sem destaque especial, lá estava ela, com as suas características verdadeiramente profissionais:

- —compatível grupo 3
- —recepção automática
- —relançamento automático de chamada
- —sinal de transmissão bem sucedida
- —corte automático do papel
- —alimentador automático de 20 páginas
- —16 tons de cinzento
- —memória para 30 números
- —captura de assinatura
- —relatória de actividade
- —telefone incorporado
- —conexão a PC por porta paralelo incorporada.

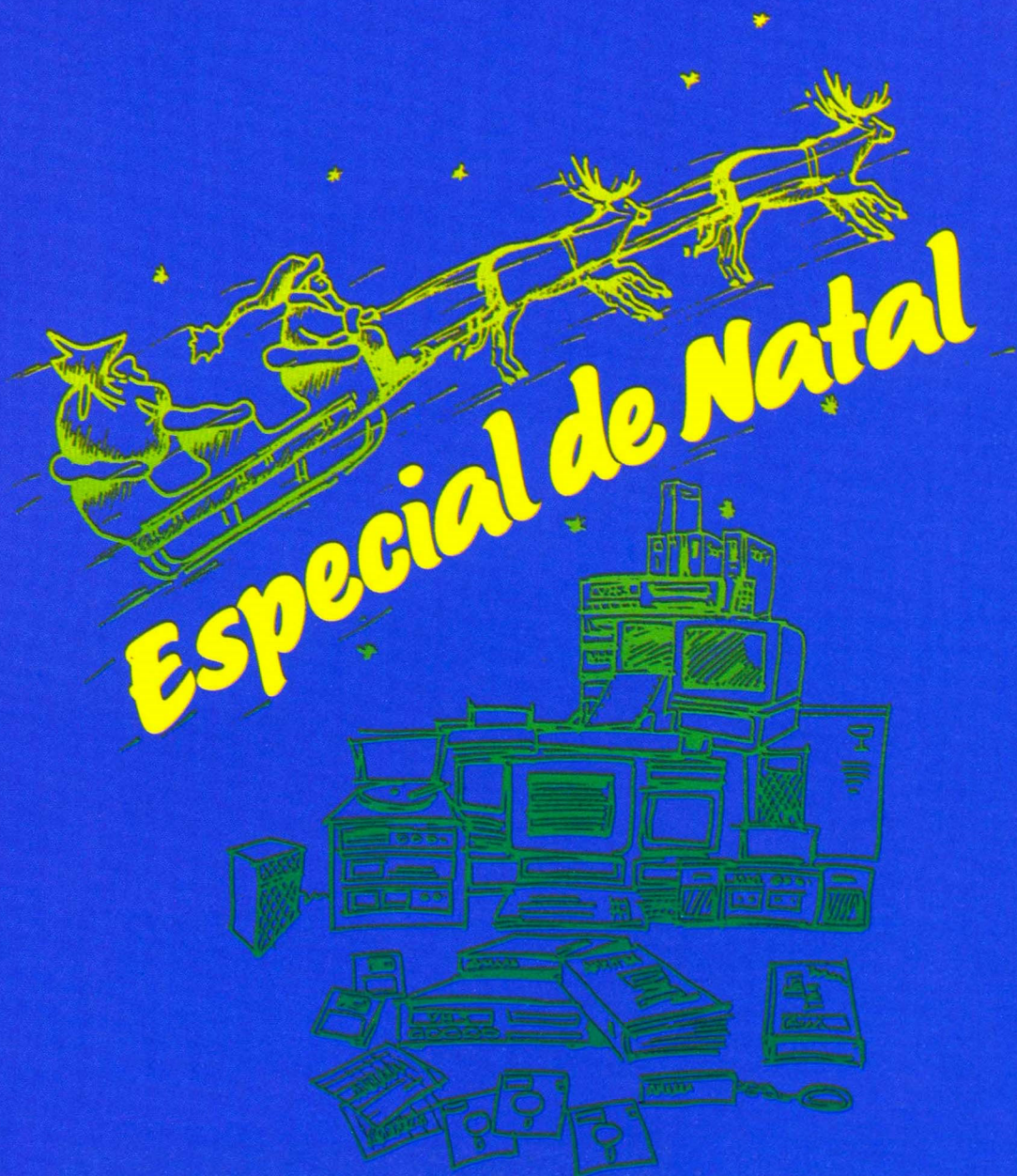
e a um preço extremamente interessante: 599 libras em Inglaterra.

CONCLUSÃO: uma visita a Personal Computer Show deixa-nos sempre a impressão de termos visitado uma feira de jogos, quer pelo tipo de expositores presentes, quer pelo ambiente que se vive na feira, quer mesmo, pela sua frequência. Num par de horas, parece-nos que já se viu tudo.

Em termos de novidades, o nosso destaque vai para o Mac portátil, para o ''Mobile Computer'' da PSION e par o fax da ''AMSTRAD'', sobre o qual falaremos com maior pormenor noutra edição da AM.

clube

AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

ESTUDO DA DECO

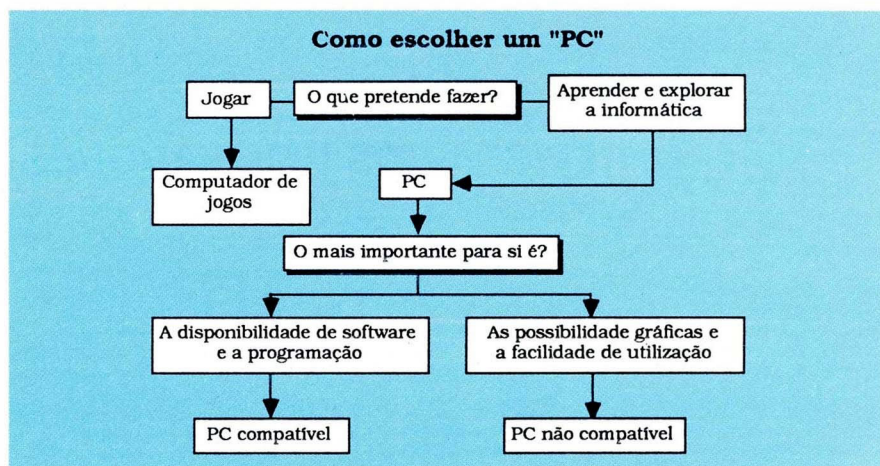
ASSOCIAÇÃO PORTUGUESA PARA A DEFESA DO CONSUMIDOR

Como escolher um computador para uso pessoal

Que tipo devo escolher?

Os computadores de uso doméstico podem ser divididos em apenas três grupos: computadores de jogos, computadores pessoais, para utilização profissional e semi-profissional, e computadores de aplicação geral, que preechem o espaço deixado pelos outros dois.

Caso pretenda utilizar um computador apenas para jogar e, eventualmente, para programar, perceber e aprender algo àcerca de computadores, ajudar nas contas da casa ou ganhar informações, então talvez um computador de jogos (não muito caro) seja o adequado. Na maior parte dos casos, os utilizadores deste tipo de computadores têm à sua disposição uma grande quantidade de jogos e outros programas simples. Por outro lado, se as suas intenções na área da informática forem suficientemente sérias ao ponto de pretender em uma máquina de aplicação profissional, poderão então optar por um computador pessoal (mais conhecido por PC de ''personal computer''). Caso necessite de trocar informações entre o seu computador e os do seu escritório ou empresa, precisará certamente de comprar um computador pessoal e de utilizar os mesmos programas, que são vendidos a um preço muito mais elevado.

Os computadores de aplicação geral igualam todos os atributos dos computadores de jogos, além de possibilitarem a utilização de programas mais evoluídos, tais como processamento de texto, gráficos e gestão doméstica. Este tipo de computadores oferecem um bom compromisso entre preço e performance, encontrando-se disponível no mercado um razoável número de programas que, a preços acessíveis, permitem satizfazer as necessidades ao nível de sistemas de informática de utilizadores individuais não muito exigentes, para uso doméstico ou profissional. Neste caso, a troca de informações com os computadores PC IBM-compatíveis (para mencionar os mais comuns) não será possível.

Os computadores aqui testados são PC/XT compatíveis, colocados no mercado por fabricantes bem conhecidos (como a IBM e a Olivetti) e por outros menos divulgados (Tulip, Victor e Zenith). Existem, contudo, dúzias de outras marcas que são, no entanto, menos relevantes em termos de vendas. Os computadores não compatíveis são todos de marcas bem conhecidas. Na maior parte dos casos, a sua performace é superior à dos PC/XT compatíveis, situando-se mais ao nível dos PC/AT compatíveis e de computadores PC ainda mais potentes. Tenha isto em conta quando comparar os preços.

Preço

Os preços dos computadores PC são extremamente variáveis (com uma tendência natural para descer), razão pela qual podem já ter mudado quando ler este teste.

Sistema operativo

O número a seguir ao nome indica a versão do sistema operativo que é fornecido com o computador (importante na compatibilização com as aplicações).

A classificação engloba a facilidade de utilização e o número de funções, tais como o controlo das unidades de discos e dos documentos arquivados. Os sistemas operativos do tipo WIMP são os mais fáceis de utilizar, mas depois de utilizar o sistema MS/DOS durante algum tempo, começa-se a aprender e a fixar os comandos necessários a um funcionamento normal.

Memória disponível

A memória utilizável de um computador é RAM (''Random Access Memory'') total disponível para fazer funcionar os programas e o sistema operativo. Para desta RAM pode ser utilizada pelo computador para comunicar com o ecrã e o teclado.

Unidades leitoras de discos

No quadro comparativo podemos encontrar a capacidade de armazenagem da unidade leitora de disquetes e do disco rígido, pois todos os fabricantes oferecem mais que uma combinação de disco (por exemplo, 2 unidades para disquetes ou uma unidade para disquetes e um disco rígido). A classificação da velocidade do disco rígido baseia-se no tempo levado pelo PC para completar tarefas que exigem muita actividade do disco, a ler/gravar programas e dados, e a funcionar com uma base de dados.

Teclado

Os computadores aqui testados apresentam nos seus teclados um número variável de teclas de funções (teclas preparadas para executar determinados comandos); alguns apresentam teclas de cursor separadas e teclado numérico separado (para facilitar a entrada de dados), aqueles que não vêm equipados com um rato têm prevista a sua integração como um extra. A facilidade de utilização dos teclados foi testada por um operador de processamento de texto, baseando-se no ''toque'' das teclas, dimensão, espaçamento e na disposição no teclado.

Monitor

A dimensão dos monitores corresponde à medida da diagonal. Para este teste utilizaram-se os modelos monocromáticos dos computadores PC compatíveis, equipados com as cartas gráficas de origem, apresentando-se, no quadro comparativo, os sistemas gráficos por elas produzidos. Dos sistemas monocromáticos, foram testados o MDA (Monochrome Display Adaptor, que apenas permite uma disposição em caracteres) e o Hercules (que produz apresentações de alta resolução, tanto em texto como em gráficos).

O sistema CGA (Colour Graphics Adaptor) pode apresentar uma baixa resolução a cores ou uma resolução média a preto-e-branco ou a uma cor, em texto e em gráficos. Os sistemas EGA (Enhanced Graphics Adaptor) e VGA (Vídeo Graphics Adaptor) produzem uma apresentação de alta resolução em texto e em gráficos, tanto a cores como a preto-e-branco, sendo ambos melhores que o sistema CGA na opção de cor. O sistema VGA é o novo ''standard'', produzindo uma resolução mais alta que as cartas EGA.

Alguns dos computadores compatíveis aqui testados vêm equipados com adaptadores dos diversos sistemas, possibilitando uma transição fácil de um sistema monocromático para um sistema a cores e vice-versa (o que é menos comun).

Cada um dos computadores não-compatíveis utiliza um sistema própio. O Apple Macintosh utiliza um monitor monocromático incorporado (e, por isso, não removível), embora estejam disponíveis outros modelos caso pretenda utilizar um sistema a cores (além de existirem diversas

COMPUTADORES PESSOAIS:Características e Resultados do Teste ©

Marca e Modelo	País de origem	Sistema operativo	Facilidade de utilização	Memória utilizável	Leitores de discos	Velocidade das disquetes	Velocidade do disco duro	Teclado: Número de teclas	Características (ver chave)	Facilidade de utilização
COMPUTADORES PESSOAIS IBM-PC/XT COMPATÍVEIS										
Amstrad PC1640 HD20MD (a)	Coreia	MS-DOS 3.2; GEM	—	640 K	5.25" 360 K/20 M	□	□	84	10F,R,N [1]	□
Commodore PC20-III	Taiwan/EUA	MS-DOS 3.21	—	640 K	5.25" 360 K/20 M	□	—	102	C,12F,N	+
Compaq Deskpro	R.Unido/EUA	MS-DOS 3.2	—	640 K	5.25" 360 K/20 M	□	□	102	C,12F,N	+
Epson PCe	Japão/Coreia/R.Unido	MS-DOS 3.2	—	640 K	5.25" 360 K/20 M	□	□	102	C,12F,N	□
Goupil G5S86	França	MS-DOS 3.21	—	640 K	5.25" 360 K/20 M	+	□	[10] 84	10F,N [1]	□
Hewlett Packard Vectra CS	França/Japão/Singap.	MS-DOS 3.2	—	640 K	5.25" 360 K/20 M	+	□	102	C,12F,N	⊞
IBM Personal System/2 Model 30	Taiwan/R.Unido	IBM PC-DOS 3.3	—	640 K	3.5" 720 K/20 M	□	□	102	C,12F,N	□
Olivetti M240 (b)	Itália	MS-DOS 3.3 [11]	—	640 K	5.25" 360 K/3.5" 720 K/20 M	+	□	102	C,12F,N	+
Samsung SPC3000 (b)	Coreia/Taiwan	MS-DOS 3.2	—	640 K	5.25" 360 K/20 M	□	□	84	10F,N [1]	□
Tandon PCX20	Áustria/EUA	MS-DOS 3.20	—	640 K	5.25" 360 K/20 M	—	—	102	C,12F,N	□
Victor VPCIIc	Japão/Taiwan	MS-DOS 3.30	—	640 K	5.25" 360 K/20 M	+	□	102	C,12F,N	□
Zenith Z-159 Model 3	Irlanda/Taiwan	MS-DOS 3.21	—	640 K	5.25" 360 K/20 M	□	□	84	10F,N [1]	□
COMPUTADORES PESSOAIS NÃO-COMPATÍVEIS										
Apple Macintosh SE20SC (b)	Irlanda/EUA	Macintosh	⊞	1 M	3.5" 800 K/20 M	[5]	[5]	105	C,15F,M,N	□
Atari Mega ST2	Coreia/Taiwan	Atari TOS; GEM	+	2 M	3.5" 720 K/20 M	[5]	[5]	95	10F,M,N	□
Commodore Amiga 2000	Taiwan/RFA	AmigaDOS	+	1 M	3.5" 880 K/20 M	[5]	[5]	94	C,10F,M,N	⊞

Teclado:
C = teclas de cursor separadas
F = número de teclas de função
R = computador com rato de origem
N = teclado numérico separado

Sistema:
C = CGA (Colour Graphics Adaptor)
E = EGA (Enhanced Graphics Adaptor)
H = Hercules
M = MDA (Monochrome Display Adaptor)
V = VGA (Video Graphics Adaptor)

[1] As teclas de controlo do cursor incorporadas no teclado numérico
[2] Não aplicável a não-compatíveis
[3] Classificação estimada (teste incompleto)
[4] Classificação estimada (computador sem adaptador de gráficos)
[5] Testes em não-compatíveis não comparáveis
[6] Computador sem adaptador de gráficos

marcas de monitores compatíveis com este sistema). Os restantes computadores não-compatíveis dependem do tipo de monitor com que estejam equipados.

A apreciação tem em conta a precisão e firmeza da imagem e a forma e o espaçamento dos caracteres.

Velocidade

A classificação apresentada para este aspecto, baseia-se no tempo que os computadores precisam para efectuar determinadas tarefas (em folhas de cálculo, processadores de texto e bases de dados) que não envolvam as unidades leitoras de discos ou qualquer outro periférico. Uma vez que o software utilizado nos computadores compatíveis não pode funcionar nos computadores não-compatíveis, não se puderam fazer testes comparativos. Contudo, experimentando tarefas equivalentes, pode-se constatar que os computadores não-compatíveis são notóriamente mais rápidos.

Som

De um computador com a classificação no quadro comparativo apenas se podem esperar alguns ''beeps'' pouco convincentes. Por outro lado, um PC classificado com está bem equipado para produzir uma grande variedade de sons (incluindo música), permitindo fazer ligações a outro equipamento de audio (como um sintetizador ou um sistema de alta-fidelidade).

Gráficos

A classificação atribuída para a qualidade baseia-se na precisão e nitidez com que os gráficos são apresentados no ecrã. A utilização do sistema corresponde à aplicação do sistema gráfico feita pelo computador por forma a tornar a sua utilização mais acessível—por exemplo, permitindo a utilização de software do tipo WIMP.

Neste caso, a classificação atribuída correspondente à facilidade de compreenção e à aplicação prática das instruções fornecidas com o computador.

Facilidade de arranque

A maior parte dos computadores testados, têm o sistema operativo instalado no disco rígido e pronto a utilizar, tornando o arranque do sistema muito mais fácil para um utilizador pouco experiente. A classificação atribuída tem em conta a facilidade de montagem total do PC, incluindo os periféricos e o sistema operativo.

Facilidade de utilização

Para classificar os computadores neste aspecto, levou-se em conta a facilida de de utilização de um PC, desde o momento em que é ligado até começar a trabalhar com uma determinada aplicação (pois a partir deste momento, a facilidade de utilização depende basicamente da própria aplicação). Todos os computadores bem classificados utilizam um sistema WIMP nas suas aplicações (e é de notar que o único computador classificado com + utiliza um sistema WIMP em todo o software).

Ruído

O ruído, no caso dos PC, tem origem no sistema de arrefecimento e no motor do disco rígido; os computadores classificados com foram considerados aceitáveis.

Monitor: Dimensão (mm)	Monitor: Sistemas (ver chave)	Monitor: Qualidade da imagem	Conectores de expansão	Interfaces (ver chave)	Velocidade	Som	Gráficos: Qualidade	Gráficos: Utilização do sistema	Instruções	Facilidade de arranque	Facilidade de utilização	Ruído	Interferências	Preço (milhares de escudos Abril/89)	APRECIAÇÃO GLOBAL	Marca
325	CEHM	□	3	J,M,P,S	+	⊖	+	□	⊞	⊞	—	⊖	⊖	304	□/+	Amstrad (a)
290	CHM	□	3	M,P,S	+	⊖	+	□	+	⊞	—	—	+	361	□	Commodore
320	HM	+	4	P,2S	□	⊖	+	□	+	⊞	—	—	⊞	527	□	Compaq
290	M	□ [4]	3	P,S	+ [3]	⊖	[6]	[6]	+	—	—	—	—	333	□	Epson
290	CEHM	+	4	M,P,S	+	⊖	+	□	+	—	—	—	⊖	463	□/+	Goupil
280	V [9]	+	5	P,S	+	⊖	—	□	+	+	—	—	⊞	637	+	Hewlett Packard
285	CE	+	3	P,S	+	⊖	+	□	+	⊞	—	□	+	433	□	IBM
295	CE	+	5	P,S	+	⊖	+	□	+	⊞	—	□	+	(b)	+	Olivetti (b)
290	CM	[7]	3	P,S	+	⊖	[7]	[7]	[7]	[7]	—	[7]	[7]	(b)	□	Samsung
335	HM	+	5	P	+	⊖	+	□	+	⊞	—	⊖	+	330	□	Tandon
330	CHM	□	2	M,P,S	+	⊖	+	□	□	⊞	—	—	—	410	□/+	Victor
285	CHM	+	4	P,S	□ [3]	⊖	+	□	+	⊞	—	—	—	339	□	Zenith
220	[2]	⊞	1	M,2S	[5]	□	+	⊞	⊞	⊞	⊞	⊖	⊞	876 (c)	⊞	Apple
295	[2]	⊞	0	J,M,P,S	[5]	⊞	⊞	⊞	+	□	+	—	—	409	+	Atari
340	[2]	+	8	J,M,P,S	[5]	⊞	□	⊞	+	⊞	+	—	□	409	+	Commodore

[7] Teste incompleto
[8] Ver texto
[9] Testado apenas com CGA e MDA
[10] Versão de 102 teclas em opção
[11] Testado com a versão 3.2

Interfaces:
J = "joystick"
R = rato
P = em paralelo
S = em série

Marcas
(a) Modelo idêntico ao SCHNEIDER PC 1640 HD 20 MD
(b) Modelo não comercializado
(c) Modelo idêntico ao SE20HD com 2M de memória

Chave do teste:
⊞ Muito bom; + Bom; □ Médio; — Medíocre; ⊖ Mau.

"CHECK LIST" DO COMPRADOR DE UM PC COMPATÍVEL

Aqui estão alguns pontos que deve verificar antes da compra (para mais detalhes veja "Compatibilidade"):

- de preferência um processador mais rápido que o antigo 8088 (8088/1, 8088-1, 8088-2, 8088-10, 8086, NEC V20 ou V30...);
- velocidade de processamento superior a 7 MHZ e, de preferência, regulável;
- mais de 640 K de memória;
- pelo menos 3 conectores de expansão livres;
- teclado completo com bloco numérico separado e teclas de cursor (ou rato);
- no caso de um monitor monocromático (a preto e branco): uma carta Hercules ou uma carta EGA, CGA ou MCGA a trabalhar em simulação;
- no caso de um monitor policromático (a cores): uma carta EGA ou uma carta MCGA ou CGA;
- dupla unidade para disquetes de 5 1/4" e de 3 1/2" e, de preferência, um disco rígido.

A Compatibilidade

Geralmente, não se podem trocar as disquetes de um computador de jogos ou de um computador de aplicação geral com as de um computador de uma marca diferente (ou para um modelo diferente da mesma marca). Para algumas máquinas nem há interesse nessa troca. Mas no caso dos PC, a troca de discos é muitas vezes necessária, particularmente quando se tem um computador em casa e outro no escritório. Esta troca é possível porque existe compatibilidade entre alguns computadores PC (normalmente IBM-compatíveis).

Algumas das áreas principais em aspectos de compatibilidade são:
A unidade central é centro de todo o sistema. No caso dos computadores pessoais compreende:

- o processador, que interpreta e executa as instruções ("Central processing Unit", CPR). Trata-se de um "chip" de silício instalado no centro do PC. Dele dependem a velocidade de trabalho da máquina e as instruções possíveis. Os processadores recentes (8088/1, 8088-1, 8088-2, 8088-10, 8086 e NEC V20 ou V30) são mais rápidas que o antigo 8088, sendo por esta razão os mais indicados. Em todo o caso, uma velocidade de trabalho de 8MHZ é perfeitamente aceitável. É de notar que existem processadores mais rápidos (o 80286 e o 80386) que equipam os PC de uma gama superior à testada (a gama AT);
- a memória central serve para guardar o sistema operativo (o "maestro" de todo o sistema) e os programas utilizados. Serve também para guardar temporariamente os dados introduzidos até à sua transferência para o material de armazenagem propriamente dito (ver mais adiante). A sua capacidade é muito importante. Esta mede-se em quilo-bytes (Kb), em que um quilo-byte comporta 1024 bytes e um byte pode representar um caracter. Digamos que uma memória de 640 K é perfeitamente aceitável para um PC;
- os conectores de expansão ("slots"). Um PC comporta uma carta principal (vulgarmente chamada "motherboard") que permite a ligação a cartas de expansão complementares e normalizadas. Estas últimas permitem uma melhor especialização do computador tendo em vista as necessidades inerentes à sua utilização principal: expansão de memória, aplicações gráficas, comunicação com outros computadores via telefone, etc. Podemos considerar como um mínimo a existência de 3 conectores livres;
- os interfaces permitem à unidade central comunicar com o material periférico (ecrã, impressora,...).

O teclado. A escolha entre uma disposição Azerty e uma disposição Querty é uma questão de habituação, embora seja possível a reprogramação de teclados (de Azerty para Qwerty e vice-versa, através da colagem de etiquetas nas teclas). Em todo o caso, quanto mais completo, isto é, quanto mais "extendido" for um teclado, melhor. Este deve incluir um teclado numérico separado e teclas de cursor, muito práticas para deslocações na horizontal e vertical do cursor de posição no ecrã.
Estas teclas de cursor são perfeitamente substituíveis por um rato

(''mouse''), que permite executar funções deslocando-se no ecrã para posições pré-definidas.

O ecrã. Existem diferentes tipos:

- o monitor monocromático, que oferece uma boa definição de imagem, muito conveniente para tratamento de texto e aplicações gráficas;
- o monitor vídeo, com um preço mais baixo que os restantes, mas com uma definição de imagem mais baixa, não deve ser considerado para o caso de computadores pessoais utilizados intensivamente;
- o monitor a cores tem igualmente uma definição de imagem mais baixa que a de um monitor monocromático, e é por esta razão que o seu uso se deve restringir a utilizações muito específicas, em que a cor é verdadeiramente necessária.

Atenção: antes da compra, deve verificar cuidadosamente se o computador está equipado com as cartas necessárias para o funcionamento correcto do monitor:

- caso se trate de um monitor monocromático (o mais comum), o computador deve estar equipado, preferencialmente, com uma carta Hercules (que oferece uma imagem de alta resolução muito nítida) ou, em útimo caso, com uma carta de cor a trabalhar em simulação preto-e-branco (CGA, EGA ou MCGA);
- no caso de um monitor a cores, ou se mais tarde pretender passar à cor, o computador deverá estar equipado, de preferência, com uma carta EGA (que também oferece uma imagem nítida e com uma boa resolução) ou, em último caso, com uma carta MCGA ou CGA (os programas escritos para a carta CGA podem ser utilizados com a carta EGA, mas o inverso não é verdadeiro);
- existem ainda outras cartas multisistemas que satisfazem os requisitos das cartas anteriores, mas com um grave aumento de preço.

Os discos. Existem dois tipos que servem de suporte à armazenagem de informação: as disquetes flexíveis (''floppy disk'') e os discos rígidos (''hard disk'').

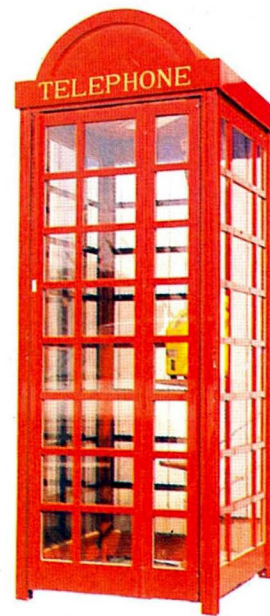
- as unidades de leitura fornecidas com os PC são normalmente para disquetes do formato 5 1/4'', com uma capacidade de 360 kb (cerca de 360 000 caracteres). Estas disquetes constituem actualmente um standard, mas as disquetes 3 1/2'', mais protegidas e menos frágeis, com uma maior capacidade de armazenagem (720 kb), estão destinadas a substituí-las a médio prazo (dentro de poucos anos). Deste modo, assegure-se antes de comprar um PC, que este vem equipado com uma unidade para disquetes de 5 1/4'', com a possibilidade de ser subistituída ou completada posteriormente com uma unidade para disquetes de 3 1/2''.
- o disco rígido está instalado dentro do módulo principal do PC. A sua velocidade de funcionamento é muito mais rápida, e a sua capacidade é muito maior (mais de 20 milhões de caracteres, comparada com a de uma disquete. O preço de uma unidade de disco rígido é elevado, mas não se trata de um luxo inútil se pensarmos no número de disquetes ocupadas pelos diferentes programas e documentos, além de tomar a sua utilização menos fastidiosa (sem as mudanças repetidas de disquetes).

O software. Existem três tipos distintos de software:

- as linguagens de programação. Com estas linguagens um utilizador pode programar o computador à medida das suas necessidades, sem precisar de comprar os programas à venda no mercado. Existem muitos tipos diferentes de linguagens, sendo o BASIC uma das mais simples;
- o sistema operativo (OS, de ''operating system''). Trata-se de uma colecção de programas que dá instruções ao computador sobre a forma de trabalhar com as aplicações (ver abaixo), além de controlar a utilização das unidades de leitura de discos e dos restantes equipamentos periféricos. O sistema operativo da IBM para os PC/XT e PC/AT é o PC/DOS (''personal Computer/Disc Operating System''). Um derivado deste sistema—o MS/DOS, desenvolvido pels Microsoft—é o sistema operativo mais utilizado pelos computadores PC compatíveis. É preciso ter em atenção que existem diferentes versões, e algumas aplicações só trabalham com uma única versão. O ano passado a IBM anunciou um novo sistema operativo chamado OS/2. Por enquanto existem poucas aplicações no mercado escritas para trabalhar especialmente em OS/2, embora a maior parte das aplicações do tipo MS/DOS funcionem com o novo sistema. Este não pode trabalhar em computadores PC/XT compatíveis (apenas em alguns PC/AT).
- as aplicações são os programas através dos quais um computador realiza as tarefas indicadas pelo utilizador. As aplicações mais comuns são o processamento de texto, a folha de cálculo e a base de dados. Existe mais software escrito para o sistema operativo MS/DOS, utilizando computadores PC/XT e PC/AT compatíveis, do que para qualquer outro. Esta é uma das principais razões para a popularidade deste tipo de computadores.

ATÉ DE UMA CABINA

PODE ENVIAR E RECEBER FAX

O NISSEI COURIER
É O FAX MAIS PE-
QUENO, MAIS LEVE
E MAIS ECONÓMICO.
E COMO É PORTÁTIL
VAI PARA TODO O LADO.

COM BATERIAS: 188.900$00
SEM BATERIAS: 174.900$00
(IVA NÃO INCLUÍDO)

DISTRIBUÍDO POR:

SOCARTEL

TEL.: PORTO: 02-9371840
LISBOA: 01-658600

O SISTEMA WIMP

Cada vez mais, os sistemas WIMP (''Windows, Icons, Mouse and Pull-down menus'', ou seja Janelas, Ícones, Rato e Menus) são utilizados pelos programadores de software na tentativa de tornar mais acessível a utilização do sistema operativo e das aplicações. A apresentação no ecrã de um sistema WIMP é feita em gráficos (a imagem é composta pelo computador em pequenos pontos)—incluindo os caracteres de texto (letras, algarismos e outros símbolos)—de tal forma que diferentes tipos de letra, estilos e tamanhos podem aparecer no ecrã quando seleccionados. Por exemplo, numa aplicação do tipo ''spreadsheet'' (folha de cálculo), a um determinado orçamento pode-se sobrepôr uma janela com um gráfico obtido a partir da informação da folha de cálculo. Essa janela pode ter a dimensão que se quiser. Pode ser colocada em qualquer posição do ecrã e pode ser ''fechada'' para se ver o que está ''por detrás''. Na maior parte dos casos, podem-se abrir várias janelas em simultâneo (fig.1).

Alguns programadores utilizam, em determinadas situações, símbolos ou pequenos desenhos para substituir palavras. Esses substitutos são vulgarmente chamados ícones. Outra característica destes sistemas é o rato: uma pequena caixa de plástico com um ou mais botões que, quando se arrasta numa superfície plana, faz deslocar o cursor no ecrã. Os botões do rato servem para seleccionar menus, ícones, janelas, zonas de texto e diversos comandos. Se preferir não utilizar o rato, as teclas de cursor continuam activas e podem servir para executar a maior parte dos comandos. Pode-se utilizar o rato para abrir menus e seleccionar, do seu conteúdo, o comando desejado. Em alternativa, pode-se escrever no teclado a sequência apropriada de letras para obter o mesmo efeito.

Este tipo de software necessita de um computador que esteja equipado com uma unidade de leitura de discos suficientemente rápida para trabalhar sem tempos de espera—por exemplo, quando se abre um menu ou se fecha uma janela. A lentidão de um computador PC/XT vulgar, a operar com um sistema WIMP, pode-se tornar verdadeiramente irritante. Assim, será necessário ter, pelo menos, um computador PC/AT compatível ou um não compatível da classe dos aqui testados.

A DISPOSIÇÃO EM CARACTERES

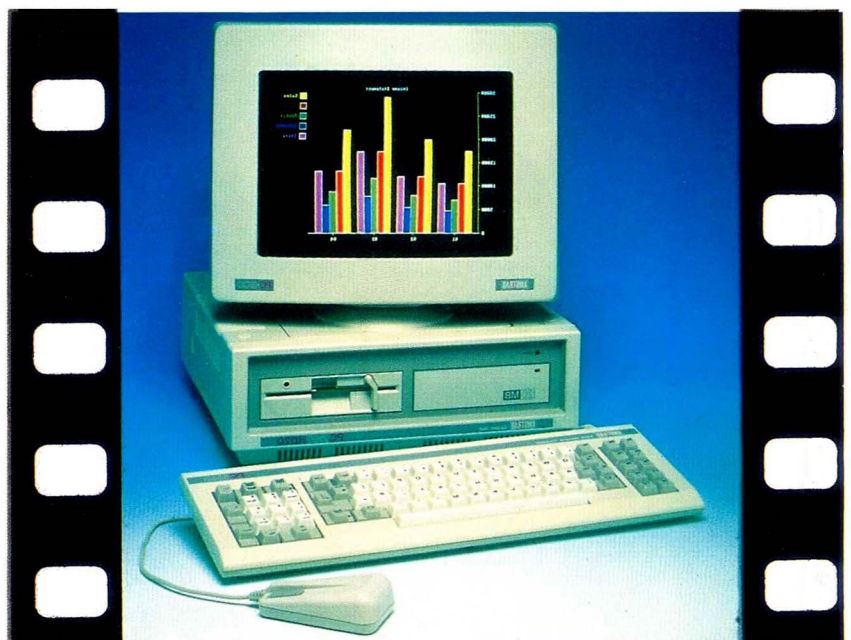
Numa disposição deste tipo, o computador pode produzir e afixar no ecrã apenas algumas centenas de caracteres—letras, algarismos e símbolos—de cada vez, tornando-se as apresentações mais rígidas e menos criativas em relação à disposição em gráficos.

Uma vez que os gráficos não são utilizados na apresentação, um gráfico não pode aparecer no ecrã em simultâneo com uma folha de cálculo, embora seja possível passar rápidamente para uma disposição em gráficos.

O menu é constituído por duas ou três linhas de instruções e comandos que aparecem no topo ou na base do ecrã. Não existem janelas, mas é possível visualizar duas ou mais áreas de uma folha de cálculo ou de um documento (através de um comando do tipo ''split screen''). No ecrã aparece apenas um tipo, estilo e tamanho de letra. Este tipo de software pode trabalhar a uma maior velocidade que o seu equivalente do tipo WIMP. Deste modo, e na maior parte dos casos, o PC mínimo necessário será um PC/XT compatível.

A ESCOLHA ACERTADA

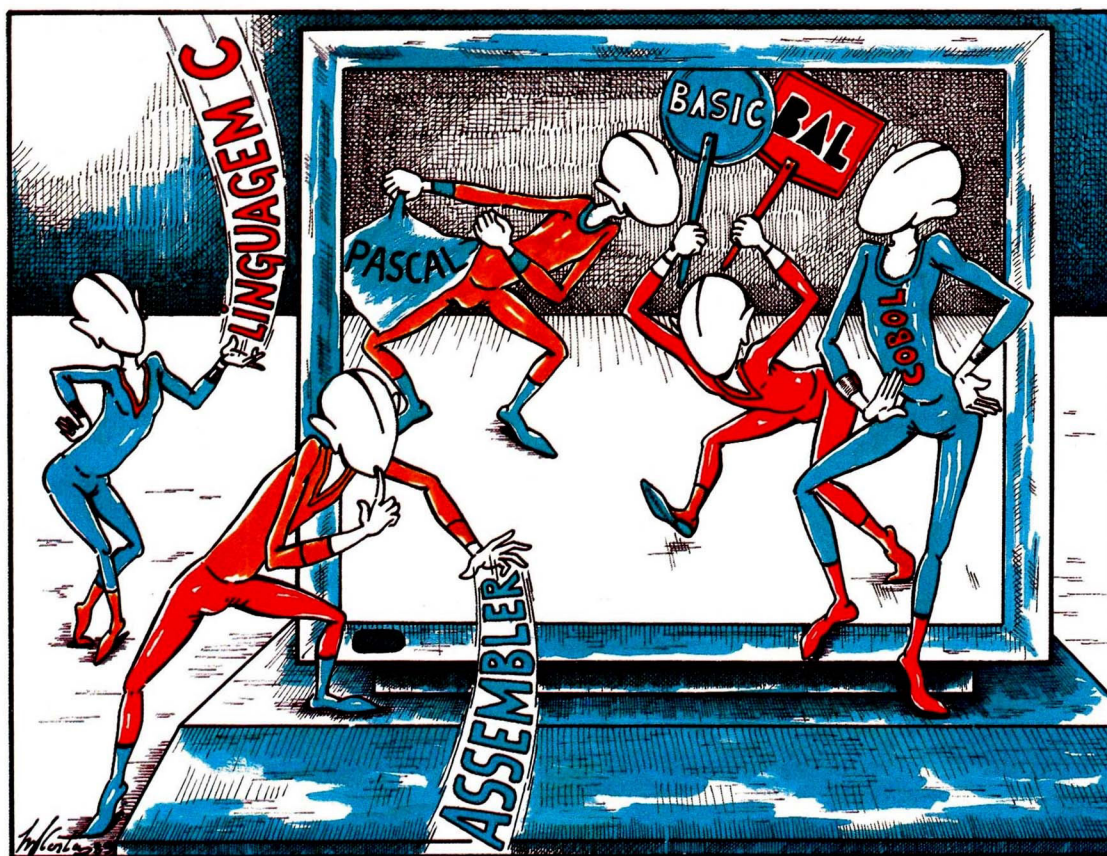
Todos os preços dos modelos indicados incluem ''drive'' de disquetes, disco rígido de 20 megabytes e monitor monocromático. Os modelos compatíveis testados têm todos uma qualidade aceitável. Destacam-se o Hewlett Packard Vectra CS e o Olivetti. **A escolha acertada recaiu no Amstrad PC 1640 HD20Mb pelo preço de venda—o mais baixo—de entre os que apresentam uma apreciação global acima da média.** Nos aparelhos não compatíveis, a qualidade geral é superior, com especial destaque para o Apple Macintosh SE20SC o melhor do teste, embora o preço seja quase proibitivo.

Testadas a 100%
mesmo em condições
desfavoráveis
de gravação

A nova
Geração de Diskettes BASF.
Ainda com mais Segurança.

As Diskettes BASF oferecem-lhe agora ainda mais segurança, também em condições desfavoráveis de gravação.

A nova geração de Diskettes BASF é o resultado do mais recente conhecimento da pesquisa e desenvolvimento, que deram origem a uma optimização de matérias, na tecnologia de produção e na segurança da qualidade.

Afinal a BASF é — o inventor do princípio da produção industrial de suportes magnéticos em geral — uma empresa leader a nível mundial nas ciências bases da Química e Física e está sempre a dar novos impulsos à evolução de modernos suportes magnéticos.

PROGRAMAS E LINGUAGENS

BASIC, PASCAL, C, ASSEMBLER... LINGUAGENS DE PROGRAMAÇÃO. ESCOLHER? PORQUÊ?

PROGRAMAS E LINGUAGENS

Basic, Pascal, C, Assembler... Linguagens de Programação. Para fazerem o quê? Um utilizador médio deve conhecer uma ou outra destas linguagens de programação? Qual o critério a seguir por um programador na escolha do utensílio de desenvolvimento que mais lhe convém?

Se você é exclusivamente utilizador, provavelmente nunca tocou numa linguagem de programação, pois os programas responderão à maioria das suas necessidades.

No entanto, abordar a programação ajuda a compreender melhor o mundo da informática e pode mesmo facilitar o diálogo com um interlocutor, aquando de uma compra.

Do ponto de vista do computador, um programa é uma sucessão de ordens (ou instruções elementares) que o computador executará mecanicamente umas a seguir às outras. Um computador não é mais que o utensílio de execução.

Mas quando, aparentemente, o computador não faz nada, ele executa um programa tão primário como perguntar repetidamente ''Alguém carregou numa tecla do teclado?''

As instruções que compõem um programa são de tal forma precisas, como por exemplo: ''inquirir o componente electrónico de gestão do teclado; o fio número 7 está alimentado em 5 volts?; se não, voltar a examinar o componente electrónico de gestão do teclado; ao contrário, se o fio está em 5 volts, então examinar o componente electrónico de recepção de caracteres; guardar em memória num determinado local os sinais eléctricos recebidos; se estes sinais são diferentes de... então voltar a examinar a gestão do teclado; se não... ''O único ponto forte de um computador é a velocidade a que ele executa estas instruções.

A todas estas instruções corresponde um código permitindo ao computador saber o que deve fazer.

A codificação destas instruções muda consoante o tipo de computador: um programa escrito para um compatível PC nunca correrá num MacIntosh e vice-versa.

O ASSEMBLER

Nos primeiros tempos da informática, os códigos eram carregados directamente em memória: constituiam séries de 0 e de 1, e pode-se imaginar o colossal trabalho que era verificar código escrito sob a forma 1001110101!

Foi então criado um utensílio com o fim de associar um código significativo a cada uma das instruções.

Por exemplo, num PC a instrução ''Ir executar o programa situado em memória no endereço 32'', traduzir-se-á por três códigos máquina ''232,32,0'' (11101000 00100000 00000000, em binário). Em Assembler podemos dar um nome ao programa do endereço 32, por exemplo, ''Imprime'' e a instrução Assembler correspondente aos nossos três códigos será ''CALL imprime'' (chama imprime).

O programador já não será confrontado com séries infernais de códigos, mas com símbolos bastante mais fáceis de memorizar, principalmente para um utilizador que saiba um mínimo de inglês: ''ADD, MUL, MOV, CALL, ...'' significam respectivamente ''somar, multiplicar, mover dados, chamar... ''O programa pode ser desenvolvido com a ajuda de um tratamento de texto. De seguida, vai ser necessário ''assemblá-lo'' com um programa que vai ler as instruções codificadas, gerando os códigos máquina ''232,32,0...''O texto codificado, introduzido por intermédio de um tratamento de texto chama-se programa fonte.

Como decerto previam, a escrita de um programa fonte em Assembler obedece a certas regras, a fim que o programa de ''assemblagem'' possa traduzir esta fonte em instruções executáveis.

É a este conjunto de regras permitindo escrever uma fonte que será compreendida por um computador que nós chamamos ''linguagem de programação''. Por abuso de linguagem, é também o utensílio que permite traduzir as fontes.

O Assembler é uma linguagem de programação de baixo nível. A cada instrução assembler corresponde um código máquina, e a cada código máquina corresponde uma instrução assembler. Cada tipo de microprocessador possui o seu próprio assembler, o que significa que esta linguagem nem sempre é semelhante de um computador para outro.

Hoje a linguagem Assembler é principalmente utilizada quando se trata de trabalhar muito próximo da máquina (examinar o estado dos diversos componentes electrónicos), para realizar módulos de cálculo cuja execução deva ser extremamente rápida, ou ainda, para desenvolver um sistema de exploração (programa sem o qual um computador não pode funcionar).

PASCAL

A linguagem Pascal foi concebida pelo professor Wirth em 1969. O seu objectivo incluía o facto de produzir um utensílio de ensino permitindo incutir aos programadores bons hábitos desde o início. Com efeito, as linguagens que já evocamos permitem muito facilmente aos programadores serem confusos e escreverem não importa o quê, não importa como nem onde.

Esta nova linguagem atingiu de tal modo o seu objectivo que se tornou simples escrever aplicações complexas resolvendo qualquer tipo de problema. Com efeito, um utensílio de programação permitindo o ensino de programação deve permitir realizar programas, e não se contentar em se divertir com alguns exemplos de escola.

Um dos conceitos sobre o qual se apoia o Pascal é a programação estruturada. O que é a programação estruturada? Significa ordem! Dito de outra forma, uma linguagem estruturada fornece os utensílios e pede a utilização desses utensílios para organizar os dados, organizar um programa em módulo para efectuar coisas bem determinadas.

O programador deve saber onde coloca o quê, o que isso representa, onde estão os programas para fazer isto ou aquilo, de forma a poder executá-los e fornecer um resultado. O Pascal fornece utensílios ajudando a desenvolvê-los para ele.

AS PRIMEIRAS LINGUAGENS EVOLUÍDAS

Os primeiros computadores destinavam-se a executar cálculo científico. De resto, a palavra inglesa ''computer'' significa calculador e designava os primeiros computadores.

Para executar cálculos, programar em Assembler não é muito prático. Sentiu-se a necessidade de poder introduzir facilmente fórmulas num programa; assim nasceu, em 1956, o tradutor de fórmula, ou ''formula translator'', o FORTRAN. O processo é similar ao do Assembler. O programador escreve um programa fonte obedecendo às regras do FORTRAN, depois fornece-o a um programa que vai traduzir esta fonte, quer directamente em código máquina, quer em fonte assembler intermédio que é necessário ''assemblar''. O programa que efectua a tradução chama-se compilador; a este compete a tarefa de traduzir as instruções FORTRAN por grupos de instrução máquina ou assembler correspondente.

Os primeiros FORTRAN possuiam ainda instruções próprias a cada computador. Entretanto, rapidamente foi sujeito a normalizações, o que permitiu transportar programas fonte de um sistema para outro com relativamente poucas modificações.

Este esforço de normalização contribuiu para que a linguagem FORTRAN fosse utilizada em muitas máquinas, o que originou o aparecimento de um grande número de programas escritos.

Uma outra linguagem foi desenvolvida nos primórdios da informática para responder a um outro tipo de problema, o da gestão. Assim nasceu, em 1959, a linguagem comum orientada para o negócio (ou gestão), o ''common business oriented language'', ou COBOL. A definição desta linguagem corresponde aos seguintes critérios: gerir bastantes dados (armazenados em discos ou bandas), gerir texto, imprimir resultados, sem necessidade de cálculos complexos.

Entre as características do COBOL, existe a utilização de palavras ignoradas pelo compilador a fim de aproximar o fonte da linguagem inglesa (desolado, mas em informática, é necessário considerar o inglês como uma língua universal), e o seccionamento do programa em quatro partes, permitindo descrever o objectivo do programa, o tipo de máquina na qual vai trabalhar, a natureza dos dados manipulados e, enfim, os tratamentos a efectuar sobre os dados.

O BASIC

Esta linguagem de programação foi concebida em 1965 para proporcionar um instrumento de programação científica e permitir uma aprendizagem rápida a engenheiros que pretendessem escrever os seus próprios programas.

BASIC significa código de instruções simbólicas para tudo fazer destinado a principiantes ou ''beginner's all purpose symbolic instructions code''.

Uma das principais diferenças entre o BASIC e as outras linguagens que referimos é o seu modo de funcionamento. Aqui, o programa fonte não é traduzido por um compilador para fornecer um executável, mas é executado linha a linha por intermédio de um interpretador.

Este ambiente de trabalho convivial permite ao programador interrogar o computador, graças a instruções BASIC, para conhecer, por exemplo, o estado das informações manipuladas pelo programa, ou ainda, permite modificá-lo facilmente sem necessidade de o recompilar.

O BASIC tem algumas semelhanças com o FORTRAN, onde foi buscar a estrutura geral, com melhorias na sintaxe e no jogo de instruções (incluindo agora o grafismo).

Relativamente às outras linguagens que acabamos de ver, o Basic é lento; com efeito, este não gera um código máquina que o computador possa executar, mas necessita da presença do interpretador para analisar cada instrução antes da sua execução.

Também, mais tarde, apareceram compiladores de Basic. Mas estes nunca poderão gerar um código tão eficiente como as linguagens concebidas para serem compiladas.

E A GESTÃO?

Acompanhamos até ao momento o aparecimento e evolução de algumas linguagens. Assim, começamos por ver o aparecimento do Assembler e, ao nível de programação-sistema, o aparecimento posterior do C. Vimos o aparecimento do FORTRAN e do BASIC e, posteriormente, o PASCAL virados para o cálculo e ensino. E a gestão? Vimos que o COBOL nasceu em 1959 e, evoluindo em várias versões, mantém-se ainda activo. Não falamos numa linguagem que se chama RPG—Report Program Generator—que tal como o nome indica, foi criada para gerar listagens de uma forma extremamente simples. Mas não existiu evolução nesta área? Na realidade existiu e existe.

Em França, no final da década de 70, apareceu o BAL—Business Application Language—Linguagem de Aplicações de Negócio, que reagrupa tudo o que para trás analisamos. Digamos que os seus criadores tiveram o cuidado de estudar as linguagens já existentes de forma a não criarem mais uma, mas uma linguagem que tivesse um pouco de todas, as partes positivas logicamente, e mais, que estivesse adaptada aos ambientes micro e mini.

Deste modo, nasceu uma linguagem criada de base para micros e não, como todas as outras, a adaptação da linguagem ao micro.

Desta forma, encontramos nesta linguagem muitas das instruções que encontramos em Basic, Pascal ou mesmo C, aliadas a instruções que permitem manipular os dados sob a forma indexada—norma ISAM—ultrapassando, nesta área, as possibilidades do próprio Cobol, já que permite ir até ao nível de organização Base de Dados relacional.

Possui uma simplicidade de sintaxe idêntica ao Basic, mas obriga a ter as variáveis e respectivas naturezas previamente definidas—Pascal, C. A riqueza da linguagem é espantosa—170 palavras reservadas—possuindo instruções que vão desde o acesso directo a um sector até à manipulação de janelas em modo gráfico, passando pela escrita e pesquisa em ficheiro Sequencial Indexado ou pelo tratamento de uma mensagem recebida numa porta de comunicações.

Tal como acreditamos que o C será a linguagem do futuro em programação-sistema, cremos que o BAL ou, melhor dizendo, o ABAL (correspondendo a evoluções sofridas pelo BAL), será a linguagem do amanhã em termos de gestão para micros, super-micros e minis, muito contribuindo para tal, a sua adopção, para os tipos de equipamentos citados, de construtores, tais como a UNISYS, a BULL e a NIXDORF.

LINGUAGEM C

Em matéria de programação, a linguagem C é um fenómeno recente, criada em 1972, mas comercializada em 1981. Trata-se da evolução de uma outra linguagem que se chamava B ou BCPL. O objectivo que presidiu à criação do C foi desenvolver uma linguagem evoluída de baixo nível, permitindo, por exemplo, escrever programas sistema , como o sistema UNIX. Por baixo nível não se deve interpretar como perjorativo, mas próximo da máquina.

Este objectivo comporta também o facto de evitar ter de efectuar desenvolvimentos longos e fastidiosos em Assembler, mas oferecer uma linguagem evoluída, quer dizer, com uma sintaxe facilmente compreensível pelo homem e independente do computador.

C é uma linguagem do tipo estruturado para permitir a organização dos dados e dos programas. Atendendo ao seu aspecto de baixo nível, dispõe de uma gama de instruções que lhe permitem explorar ao máximo as possibilidades de uma máquina, da sua unidade de cálculo, da sua memória e permitindo também, a quem o deseje, dialogar com a electrónica.

Uma linguagem que permite fazer de forma óptima quase que não importa o quê. No plano de eficiência, estima-se que um programa em C bem codificado atinja 90% da eficiência de um programa bem codificado em Assembler. O conhecimento da máquina não é necessário para programar em C, mas um programador Assembler pode saborear a forma como o programa explora o computador em confronto com as outras linguagens.

Os critérios que permitiram definir esta linguagem são únicos e responderam às necessidades patenteadas por muitos programadores-sistema; ora, como são os programadores-sistema que permitem a um computador funcionar, rapidamente se disponibilizou a linguagem C um pouco por todas as máquinas. A nível de ''sistema'', poderá ser a linguagem do amanhã.

No entanto, trata-se, devido à sua riqueza, de uma linguagem de programação de manipulação e utilização mais complexa do que qualquer outra. O nosso conselho, se se interessa pela programação, seria de começar pelo Pascal e depois—uma vez familiarizado com os princípos da programação—de evoluir para a C se o coração assim vos ditar, ou então explorar novas linguagens, ou ainda permanecer no Pascal.

Exemplo de um programa BAL

```
; Este programa escreve no écrã
; 10 vezes a string ''Amstrad Magazine'',
; precedida pela numeração da linha
; A instrução ''program'' seguida de um nome,
; indica o início do programa
; REM ou ; ou * significa comentário
Program ''Amstrad''
dcl4c+          ; Declaração da variável i
segment o       ; Início segmento
for i=1 to 10
        print=1:tab(01,i),(nz),i,(w16), ''Amstrad Magazine''
next i
eseg 0          ; Fim segmento
end             ; Fim programa
```

INTELIGÊNCIA ARTIFICIAL

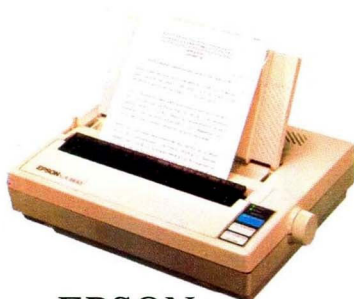
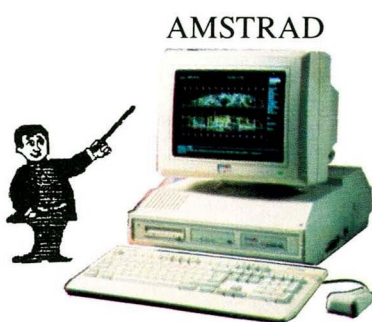
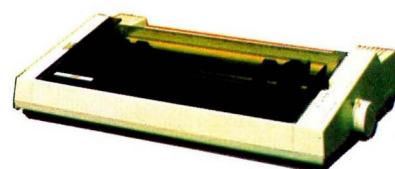
Não entenda pelo título acima que um computador possa tornar-se inteligente. Isso não é mais que um mito alimentado por alguns filmes de ficção científica, e por algumas pessoas que a todo o custo nos querem fazer crer que o homem mais não é que um bocado de carne.

Não é por acaso que o Larrouse da informática não hesita em qualificar esta expressão como ''abuso da linguagem''. A única inteligência existente num programa é a que um programador quis aí colocar, e mesmo assim, mais não se trata de uma inteligência aparente, com a visualização da palavra ''thinking'' (estou a pensar) no momento adequado.

Propomo-nos aqui definir inteligência artificial como ''a matrização de um certo número de comportamentos observados no homem e que podem ser mecanizados''.

Assim, podemo-nos servir de um computador para acelerar processos de cálculo ou pesquisas em bancos de dados; este não tem que fazer mais do que isto, ele pode fazê-lo muito depressa e muito bem.

Citemos aqui uma linguagem de programação surgida das técnicas da inteligência artificial Prolog (não confundir com PROLOGUE—Sistema Operativo multiposto de origem francesa destinado a micros) ou a programação lógica. O princípio deste tipo de linguagem é fundamentalmente diferente das outras. Aqui não se descreve ao computador o que ele deve fazer, mas descreve-se-lhe os dados e as informações intervindo no problema. Estes dados podem ser memorizados para constituirem o que se chama uma base de conhecimento.

Uma vez esta base de conhecimento construída, podemos interrogar para obter resposta a uma questão. O controlo da máquina passa para um mecanismo que examina a base de conhecimento para determinar os factores desconhecidos expressos na questão. Poder-se-ia dizer que este mecanismo, o motor de inferência, examina todas as possibilidades possíveis a partir dos elementos da questão para determinar quais os elementos presentes na questão descritos como desconhecidos.

A GESTÃO DE BASE DE DADOS

Na micro-informática moderna apareceram programas de gestão de base de dados. Estes permitem num quadro de projecto de gestão organizar um conjunto grande de informações e de dados num sistema informático.

Estes gestionários de base de dados também possuem comandos para permitir interrogar a base de dados, ou para que um gestor possa dialogar com esta.

Entre os primeiros programas de gestão de base de dados para micros devemos citar DBASE, hoje DBASE III PLUS e DBASE IV. Esta base de dados oferece a possibilidade de armazenar séries de comandos em ficheiros para, posteriormente, serem executados: é a programação. Ao escrever-se um programa para DBASE recorre-se a algumas instruções do tipo estruturado. Tendo sido um dos primeiros programas neste domínio, DBASE foi largamente divulgado e constitui um standard nesta área. Entretanto apareceram outros produtos no mercado utilizando a linguagem e a sintaxe de DBASE e permitindo também programas escritos nessa linguagem. A presença do gestionário da base de dados não é necessária.

Uma linguagem como a de DBASE é facilmente acessível ao novo programador, e permite realizar de uma forma rápida e fácil programas no domínio da gestão com um pouco menos de conhecimentos informáticos que os necessários para outras linguagens; pode ser outro método de aprender a programar.

Assinalemos que o mundo das bases de dados está em plena evolução e que produtos novos como Paradox3, permitem realizar uma quantidade enorme de coisas sem necessidade de tocar numa única linha de programa.

É possível forjar écrãs, validações, elos entre ficheiros à medida que se vai utilizando o Paradox, e isto de forma suave e interactiva. Juntemos a isto um sistema poderoso de interrogação de base de dados (QBE: Query by Example; quer dizer, Interrogação por Exemplos).

Citando Base de Dados, falemos um pouco sobre SQL (Search and Query Language, linguagem de inter-

GESTÃO DE BASES DE DADOS

rogação e de pedidos). Isto não é uma linguagem de programação, nem uma linguagem destinada ao utilizador de base de dados. Trata-se de um utensílio, transparente a vários níveis, permitindo interrogar uma base de dados situada, por exemplo, noutro computador.

Assim DBASE IV E PARADOX possuem um módulo SQL que lhes permite, sem que o utilizador dê por isso, obter e gerir dados que não lhe pertencem e que estão situados num mini-computador localizado a milhares de quilómetros daqui. Estes dados apresentar-se-ão como dados DBASE ou PARADOX. Dito de outra forma, SQL é utilizado para gerir o diálogo entre um computador e uma base de dados.

Antes de concluirmos sobre programação, e as linguagens de programação, devemos mencionar todos esses programas dos nossos dias, ditos de burótica, existentes no mundo da micro-informática. Encontramos base de dados, folhas de cálculo, tratamentos de texto, gestão de impressão... todos estes programas possuem em maior ou menor escala a possibilidade de memorizar operações, ou possibilidades de cálculo, e isto... ainda é programação!

EM SÉRIE

Um computador é constituido por três componentes de HARDWARE: a unidade central de processamento, a memória e a entrada/saída. É através da entrada/saída que a unidade central de processamento comunica com o exterior. Existem duas maneiras de implementar essa comunicação: em paralelo e em série. É precisamente sobre entrada/saída série e comunicações a ela normalmente associadas que pretendo escrever neste e noutros artigos sob este mesmo título.

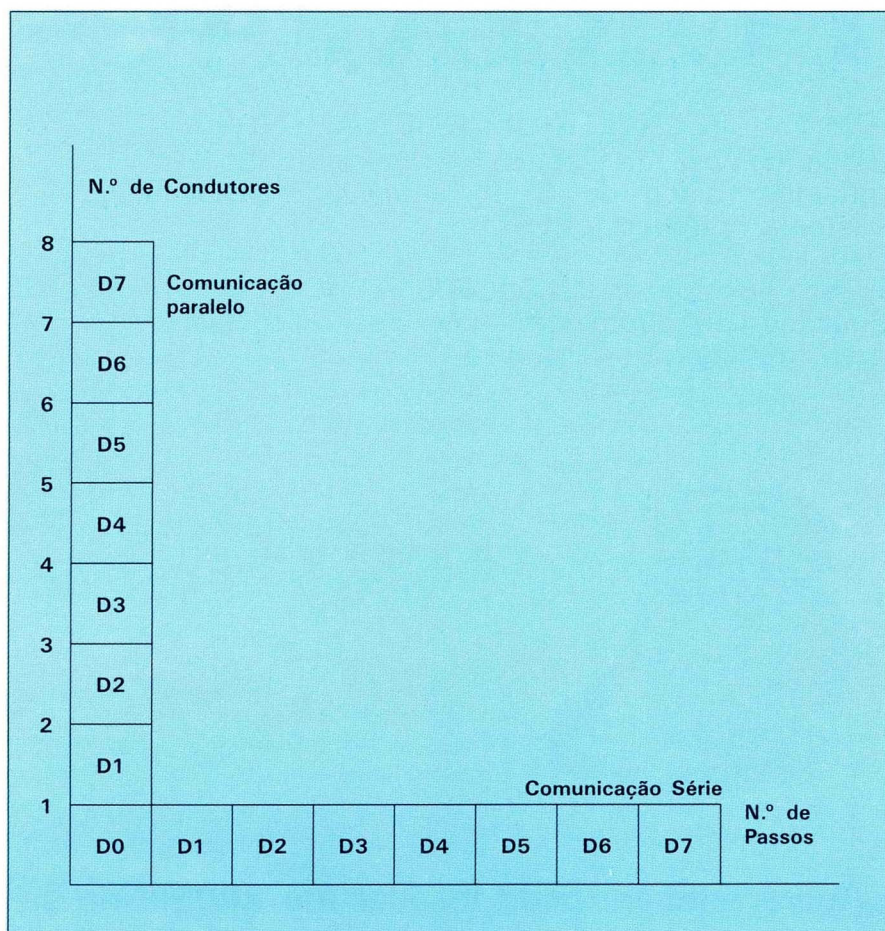
SÉRIE VERSUS PARALELO

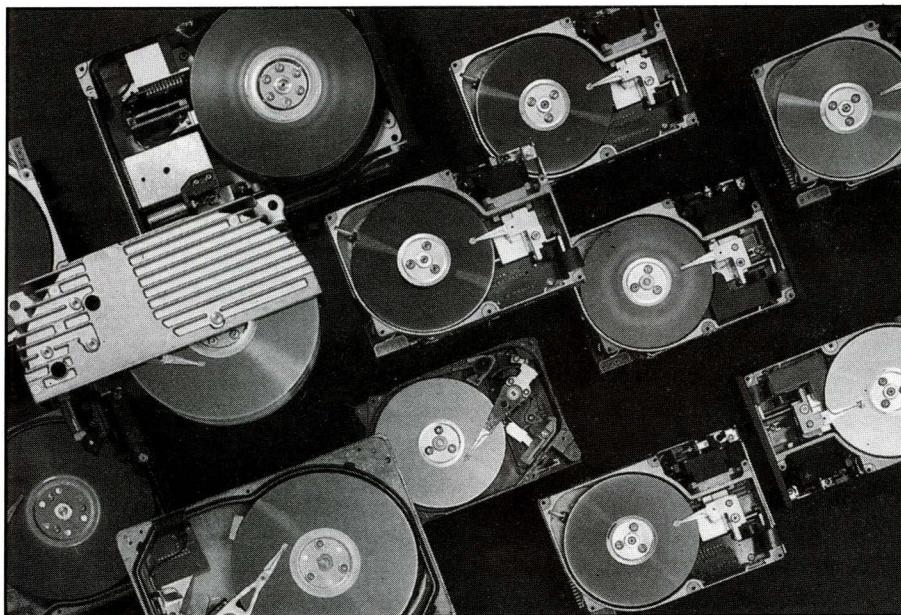
A unidade de comunicação em informática mais ususal é o byte (conjunto de 8 bits). Para transferir 1 byte de informação, ou se utiliza um conjunto de oito linhas (uma linha por bit) e transfere-se essa informação num passo—comunicação paralela—, ou se utiliza uma única linha e transfere-se o byte em questão em oito passos—comunicação série. (ver figura 1)

A comunicação paralela é fácil de compreender (diria mesmo é intuitiva), simples de implementar e utilizar e permite uma alta velocidade de transferência de informação. Então... porquê falar de comunicação série?A principal razão é: a comunicação paralela utiliza oito linhas para transferir um byte de informação... num sentido. Se precisarmos de uma comunicação bidireccional temos necessidade de mais oito, mais uma linha de referência dos zero Volt, pelo menos, enquanto que a mesma comunicação em série se resume à utilização de três linhas apenas.

Evidentemente que quando se pretende estabelecer uma comunicação entre o computador e um periférico que deste diste alguns metros esta vantagem não tem qualquer significado. Contudo, se essa distância for de algumas dezenas de metros, ou mesmo centenas, já é razoável optar pela comunicação série. Mas se pensarmos que grande parte das comunicações se fazem a quilómetros de distância e, principalmente, que essas comunicações se fazem utilizando as linhas telefónicas—já nem sequer se trata de uma questão de opção—então as comunicações série têm absoluta exclusividade.

Posto isto, vamos então debruçar-nos sobre a comunicação série em si. Como atrás foi dito, a comunicação série resume-se, grosso modo, a enviar informação por uma única linha fazendo variar, de acordo com essa informação, o potencial dessa linha, ao longo do tempo. Mas, como é evidente, essa variação não pode ser qualquer: tem de obedecer a determinadas leis. Uma é a contagem do tempo, (ou seja, a cadência de sucessão dos bits—chamado baude rate) feita por um sinal de relógio (clock), entre outras. É fundamental que estas leis sejam as mesmas na entidade que envia informação e na entidade que a recebe (ver figura 2).

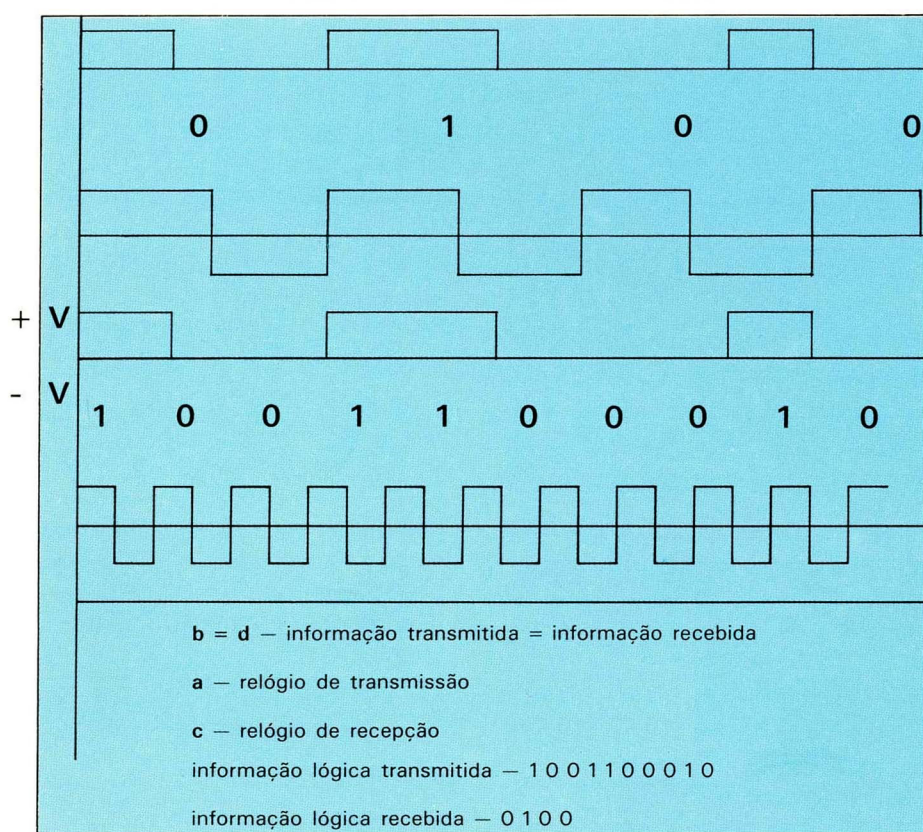
Assim diria que comunicar em série é, fundamentalmente, codificar e descodificar informação.

ENGENHARIA INDUSTRIAL & PLANEAMENTO ELECTRÓNICO, LDA.
Rua Almeida e Sousa, 33, r/c Esq.
Telefs. 69 03 18 - 68 12 43 — Telex 65701 - P — Fax n.º 67 85 43
1300 LISBOA - PORTUGAL

A NOSSA EXPERIÊNCIA É A SUA SEGURANÇA

TODO O TIPO DE DISCOS DUROS E "FILE CARDS" PARA O SEU COMPUTADOR

PORTAPAC

UM NOVO CONCEITO DE DISCO AMOVÍVEL
Pode colocar no espaço do Drive de 5′ ¼ qualquer disco de 3′ ½, amovível com o sistema Portapac.

E isto por apenas 20 000$00 + IVA.
Incluindo armadura com chave e saco especial para transportar o disco.

Agora também pode instalar até 200Mb no seu Amstrad sem qualquer problema de consumo pois os DISCOS CONNER consomem apenas cerca de 2 W cada.

Há várias formas de codificação. Antes de as analisar vamos tecer algumas considerações sobre o meio da comunicação, ou seja, a linha que lhe serve de suporte, nomeadamente no que respeita à banda passante e ao acoplamento.

Banda passante é a gama de frequências que se podem fazer passar por uma linha sem haver distorção significativa. Isto tem a ver, fundamentalmente, com a cadência de transmissão de informação, ou seja, o número de bits que podem ser enviados numa unidade de tempo. Quanto maior for a banda passante maior poderá ser essa cadência. Contudo, esta relação não é a mesma para todas as codificações, ou seja, uma determinada banda passante, a uma determinada cadência. Porém, se se usar uma outra codificação essa mesma informação poderá ser enviada a uma cadência maior como adiante se verá. De qualquer forma, toda a análise se fará, como é vulgar neste tipo de considerações, para o caso mais desfavorável, ou seja para a frequência máxima (fundamental) a enviar pela linha.

O acoplamento prende-se com o modo como os sinais são transmitidos ao longo da linha e de equipamento para equipamento e, basicamente, são de dois tipos: ac e dc, conforme se podem utilizar sistemas de acoplamento com isolamento ou se necessitar acoplamentos sem isolamento (com a mesma referência ao zero volt).

Ao analisarmos as diferentes codificações faremos ainda referência à imunidade ao ruído (ou seja, ao nível que uma interferência pode atingir sem prejudicar a fidelidade da informação veiculada).

Assim, para concluir estas considerações iniciais, deve-se usar a codificação que melhor aproveite a banda passante da linha, utilizando ao mesmo tempo o acloplamento mais eficiente e barato (dado estarem condições económicas na origem da transmissão série) e com maior imunidade ao ruído. Complementarmente deve fornecer um mecanismo eficaz de detecção de erros de transmissão (eventualmente de correcção mesmo).

Inicialmente vamos dividir os tipos de codificação em duas classes

—NRZ (sem retorno a zero)
—RTZ (com retorno a zero)

definidas da seguinte forma:

NRZ (do inglês ''non return to zero'')—A transição de níveis faz-se no início do ciclo de relógio e o sinal mantém-se nesse nível durante todo o período.

RTZ (do inglês ''return to zero'') —Há uma transição do nível lógico que define o estado (zero ou um) para o nível zero, a meio do bit, pelo menos para um dos estados lógicos. Ou seja, o sinal está em zero em, pelo menos, metade do ciclo.

Em qualquer destas classes os valores lógicos podem ser representados por níveis eléctricos ou por transições. A representação dos níveis pode ainda ser unipolar ou bipolar, tal como a representação por transições, podendo ainda esta última representar um valor lógico pela existência ou ausência de transição de menos para mais e vice-versa.

Na breve análise de frequências que faremos, teremos em conta apenas, tal como atrás foi dito, a banda passante até à frequência fundamental máxima do sinal codificado.

—NRZ

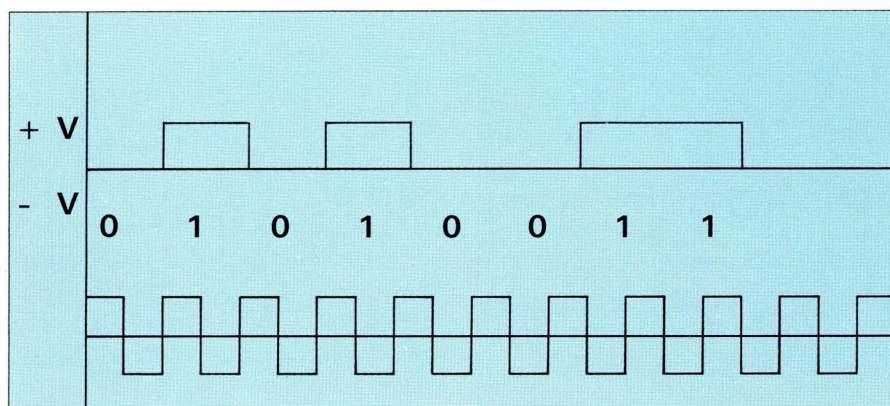
I—A primeira codificação a analisar é a unipolar NRZ. Os uns são representados pelo nível alto e os zeros pelo nível zero.

fr é a frequência do relógio

—Qualquer transmissão contém sempre a componente contínua (o que impede o acoplamento por transformador).

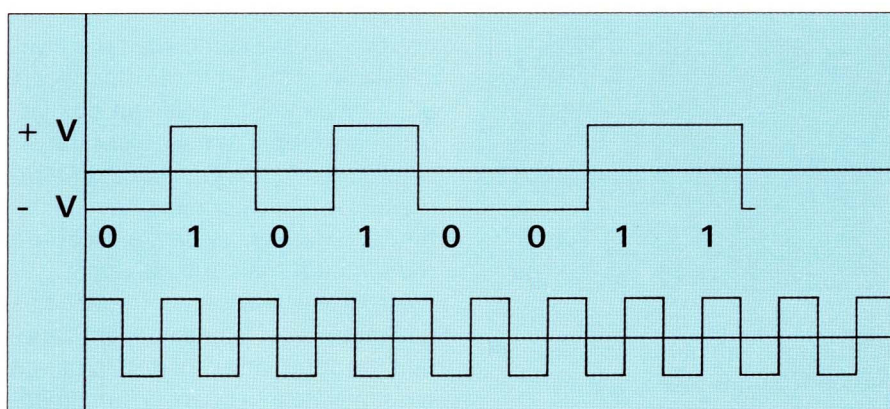
—A frequência fundamental máxima é fr/2.

II—Desta deriva a bipolar NRZ (se os níveis forem +12V e -12V é o tipo de codificação utilizado pelo Standard RS232C que mais tarde veremos com mais pormenor). O valor um é representado por uma tensão positiva e o valor zero por uma tensão negativa.

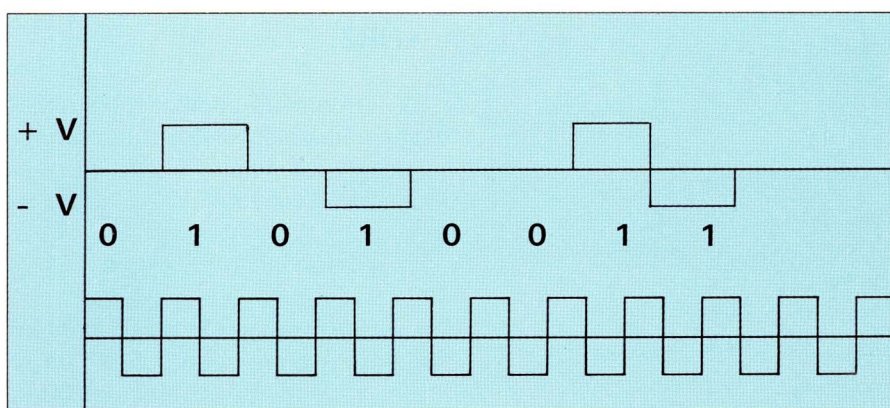
* Frequência fundamental máxima é fr/2

—Nesta a componente contínua diminui (chegando a ser zero à frequência fundamental máxima, isto é, na transmissão alternada de zeros e uns). Contudo tal é insuficiente para garantir um bom acoplamento a.c.

—Dados os níveis de sinal tem uma maior unidade ao ruído.

III—NRZ Bipolar alternada.

Um outro tipo NRZ bipolar é o que utiliza para o valor zero a tensão nula e para o valor um, alternadamente, tensão positiva e tensão negativa.

IV—NRZ Diferencial

A transição no início de um ciclo de relógio equivale ao valor lógico um e a não transição equivale ao valor lógico zero.

Estes dois últimos não trazem vantagens visíveis relativamente aos dois primeiros.

—RTZ

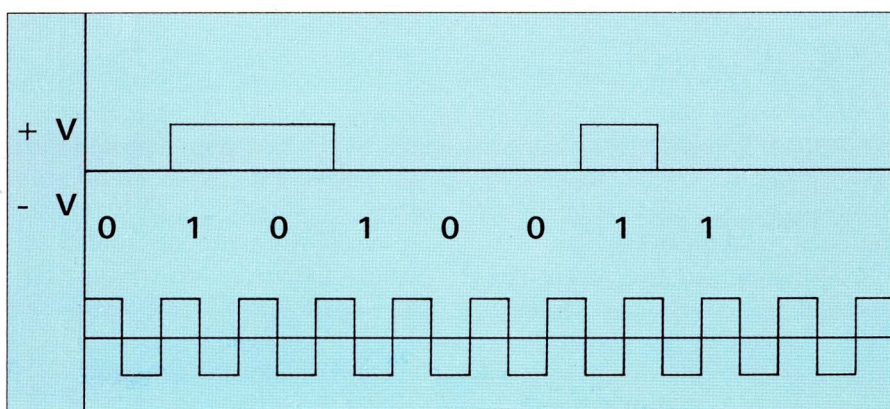
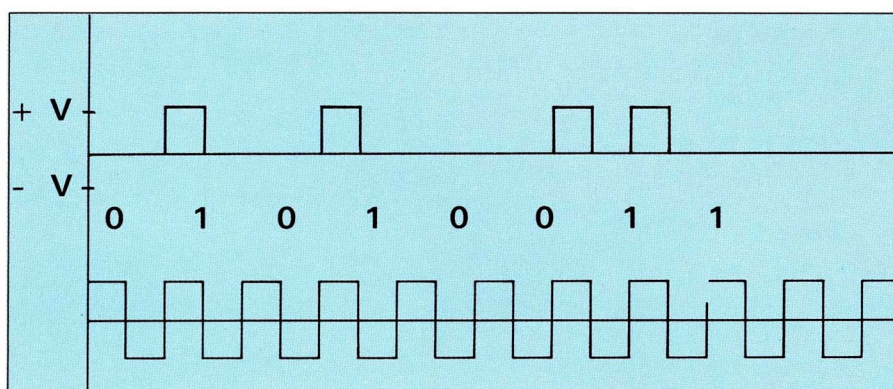
I—UNIPOLAR

Um representado com tensão positiva, com retorno a zero a meio do ciclo. Zero, tensão zero.

A frequência fundamental varia desde a componente contínua (informação a zeros) até à frequência de relógio (informação a uns).

II—BIPOLAR

Um representado pela tensão positiva, o zero pela tensão negativa com retorno a zero.

A frequência fundamental é a frequência de relógio (informação toda a uns ou a zeros). Contudo existe e é sempre transmitida a frequência de relógio o que pode ter grandes vantagens, por exemplo, para gravação em fita magnética.

III—BIPOLAR ALTERNADA
''0''—zero volts
''1''—Alternadamente +V e -V

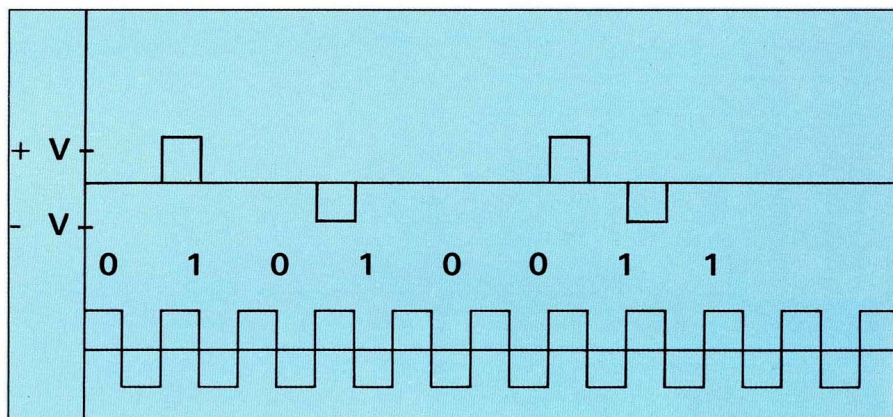
Relativamente à anterior diminui a componente contínua, mas não garante a frequência de relógio.

IV—''DELAY MODULATION''

O ''1'' representado por uma transição a meio do bit e o ''0'' sem transição, excepto se se seguir um outro zero em que a transição se faz no fim do bit.

V—''PAIR MODULATION''

Modulação bipolar em que os bits são codificados dois a dois e que traz em si já uma codificação possível de detecção de erros, dado que, para as quatro codificações possíveis de informação de dois bits existem nove codificações possíveis da modulação, sendo portanto, 5 dessas codificações sinónimo de erro de transmissão. Existem dois modos (Modo 1 e Modo 2).

NOTA: De notar que as combinações + +, — — e 00 (possíveis de serem o resultado de curto circuito da linha a qualquer uma das tensões ou do corte desta) são combinações de erro.

VI—BIFASE (ou codificação Mancheter II)

A frequência fundamental varia entre a metade de frequência de relógio e a frequência de relógio.

De todas as codificações RTZ a mais importante é a bifase (Manchester II). Assim, a jeito de conclusão vamos compará-la com as codificações NRZ, duma forma genérica.

—NRZ

—As codificações NRZ são simples e práticas pois não necessitam de dispositivos de codificação e descodificação.

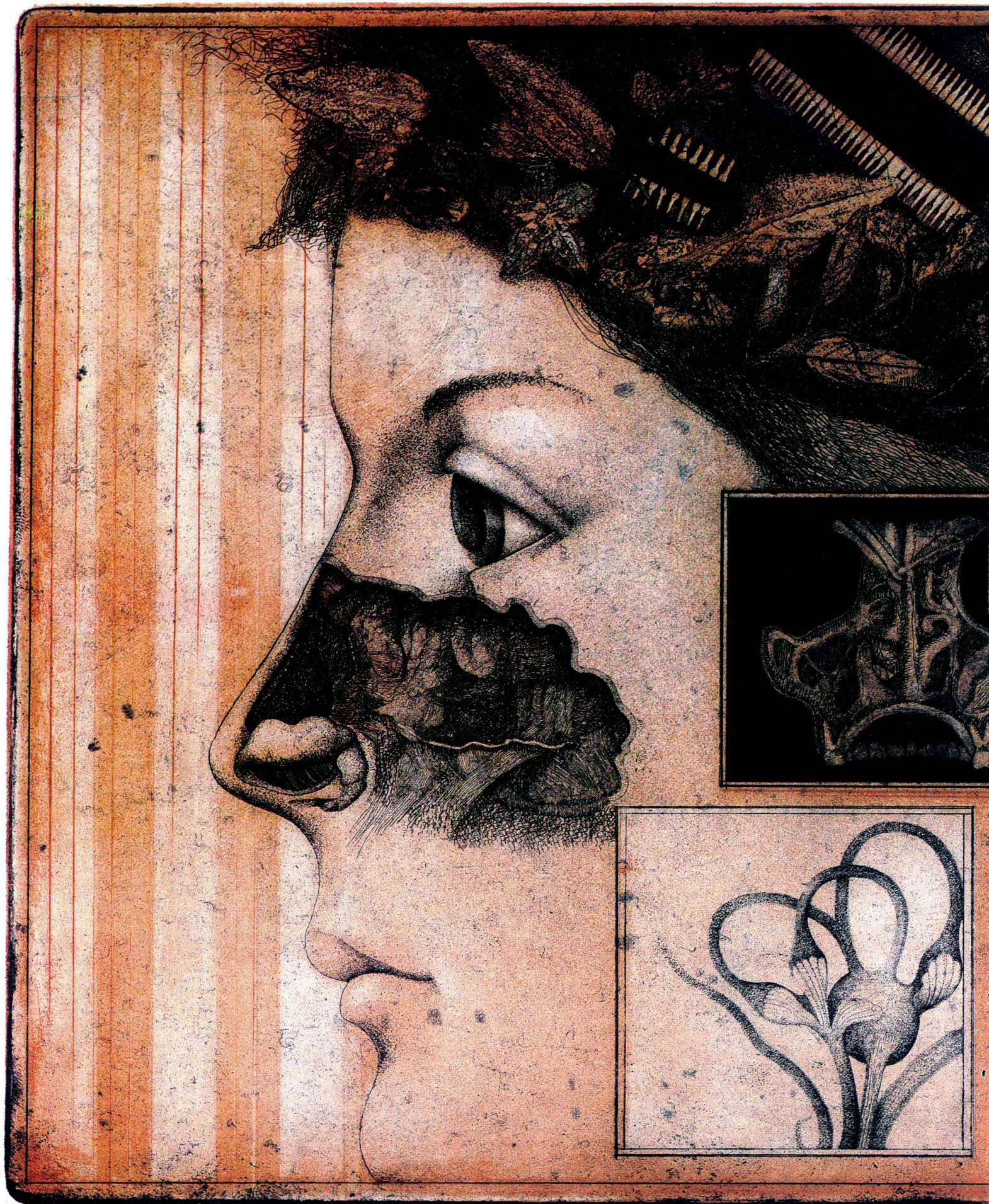
—Fazem uma óptima utilização da largura da banda pois, como se sabe, um canal de transmissão é um filtro passa baixo e a frequência fundamental da codificação NRZ varia desde a componente contínua até à metade da frequência de relógio.

—No caso de serem necessários amplificadores intermédios a sua largura de banda necessária é inferior pelo que ficam mais baratos.

—São muito mais expandidas (note-se que o conhecidíssimo RS232C pertence a esta classe de codificações).

MANCHESTER

—Já não é argumento de peso a complexidade (e preço) da codificação pois, inclusivé, já existem com-

Reprodução de gravura a água-forte de José Faria

OLFACTO

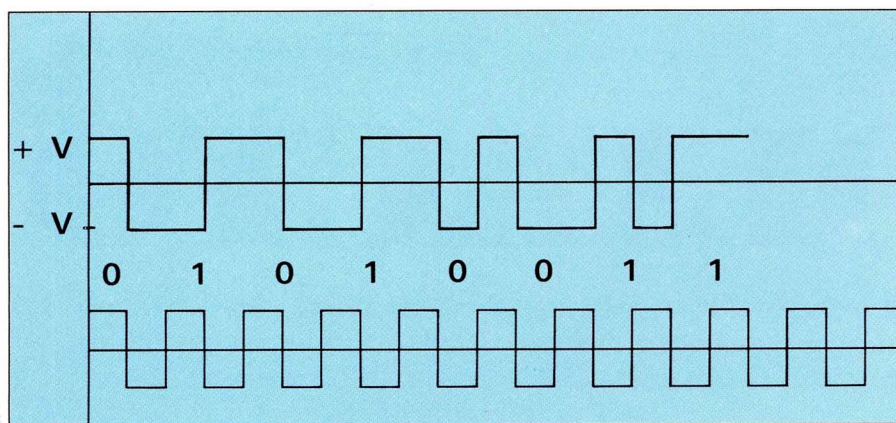
O olfacto é a inspiração decisiva que desperta no Homem o sentido das coisas. Um radar especial que capta o fluxo etéreo da vida, talvez mesmo a sua essência mais pura.

A Sopsi encara o mundo sensível como a lufada de ar fresco de que a informática necessita para florescer em pleno. Impregnando a sua influência em múltiplas áreas do ramo, em inúmeras empresas, a Sopsi liberta assim um perfume sedutor, distinto, envolvente.

Quando se tem o faro apurado, o sentido de orientação não é volátil. Hoje, a SOPSI representa e comercializa das melhores marcas e produtos do mercado. É maioritária na Cominfor, Publinfor, Amsónica e Socartel. Formou a Lusicomp, uma empresa retalhista profissional de características únicas. Penetrou no ramo das telecomunicações com a Telefónica e autonomizou a prestação de serviços de "hardware" com a criação da Scati. Materializou a entrada da informática nas artes gráficas, sendo maioritária na Tipografia Guimarães. E vê-se já colocada como a quarta maior empresa de informática no País — a segunda com capital integralmente português —, depois de ocupar em 1985 a décima-nona posição.

Já lá vai o tempo em que o Homem se contentava apenas com incenso e mirra. O reino dos odores é agora muito mais exigente. E o da informática, também. Mais do que nunca, a personalidade jovem e dinâmica da Sopsi sente-se no ar. Tal como um cheiro a novo, inebriante, afirmativo.

O Sentido da Informática

Sopsi - SOCIEDADE PORTUGUESA DE SISTEMAS DE INFORMÁTICA, S.A.
AV. DA BOAVISTA, 2881-2.º — 4100 PORTO

ponentes (por exemplo o MED da HARRIS Semicondutor) que fazem a conversão Manchester II—NRZ e NRZ—Manchester II.

—Realmente a frequência fundamental de uma transmissão Manchester II varia entre fr/2 e fr. É, portanto, mais atenuada pela linha de transmissão. Contudo, a codificação Manchester não tem componente contínua o que facilita o acoplamento A.C. que é mais fácil, eficiente e barato que o acoplamento D.C..

—Mesmo precisando de possíveis amplificadores com uma largura de banda superior (não parece um factor crítico quando esta frequência se situa na gama das dezenas de Kilohertz) esta amplificação é mais simples em A. C. do que D. C., e faz-se com maior imunidade ao ruído pois este é superior às baixas frequências (dado que as suas principais causas na amplificação são o ''offset'' e o seu ''drift'').

Também ao nível da transmissão o ruído é maior às baixas frequências.

Acresce ainda que as limitações de frequência nos canais de transmissão são cada vez menores (utilização de cabos coaxiais e fibras ópticas). Esta codificação, contudo, é ideal para gravação em banda magnética por esta não admitir uma componente contínua em baixas frequências e, ainda, por a codificação Manchester II ser ''self-clocking'', ou seja, contém em si o sinal de relógio. (Há sempre uma variação em cada período de relógio).

—Note-se que uma das dificuldades na gravação em banda magnética é a manutenção da velocidade de arrasto constante. Para se referenciar a uma frequência de relógio, ou se grava essa mesma frequência num canal auxiliar ou então ...utiliza-se uma codificação ''self-clocking''!

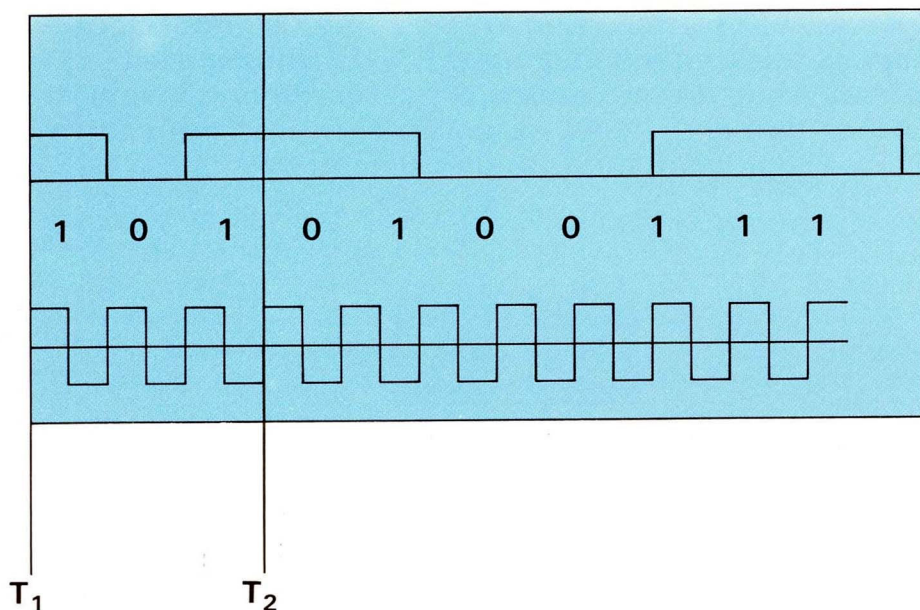
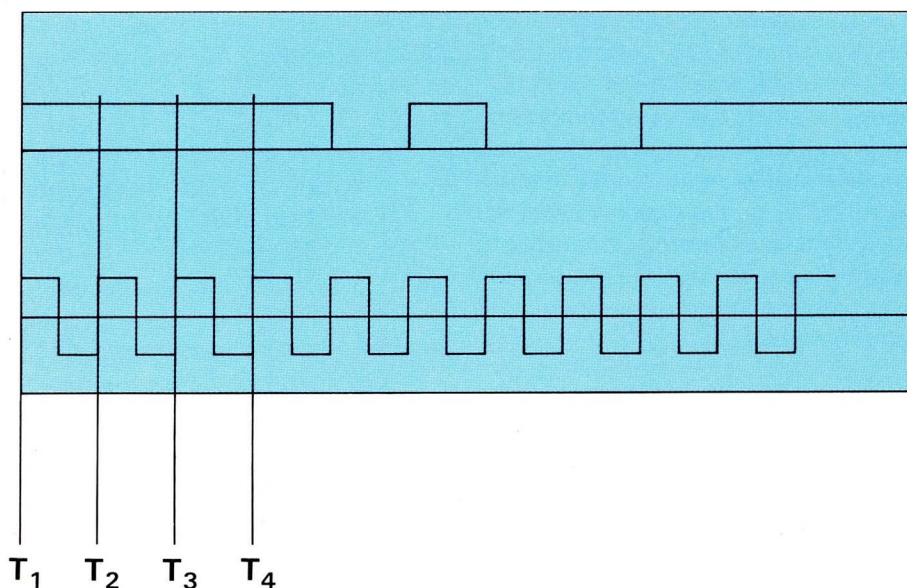
Esta característica já não se revela tão importante em transmissões por fios, pois as variações de clock podem ser superados pela geração local de um clock 16, 32 e 64 vezes superiores à frequência fundamental de transmissão, como aliás, veremos quando nos referirmos ao RS232C.

Vimos já na figura 2, que é condição essencial para que dois equipamentos se entendam que ambos tenham o mesmo relógio (ou um relógio igual). Mas isto não basta. Vejamos a figura 19

Se o equipamento receptor ''entender'' que o byte enviado pela linha começa no momento T1, lerá a informação de 8 bits 1011 1000, que é radicalmente diferente do byte lido, considerando que este começa no instante T2 e que seria 1100 0111.

Bom, poderá dizer-se que é evidente que o byte começa no tempo T1... ou que bastaria saber qual é o primeiro bit e, depois, contar sempre oito bits para o início do byte seguinte. Acontece porém, que nem sempre há informação para enviar, pelo que, entre mensagens, a linha está em estado de repouso. Vejamos que aqui é mais difícil de saber onde começa um byte (suposto isolado) e onde acaba, desde que tenha algum bit de início ou de fim igual ao estado de repouso. Fig.20

Assim a comunicação transmitida poderá ser entendida como começando no instante T1 (1111 0100), T2 (1110 1001), T3 (1101 0011) etc. sem que seja evidente qual o momento que deve ser considerado para início de informação.

Assim, vemos que não basta (embora seja absolutamente necessário) que o relógio de transmissão e de recepção sejam iguais. É necessário

também que estes estejam sincronizados. E isto pode ser feito de dois modos: Assíncrono e Síncrono.

MODO ASSÍNCRONO

Existe sempre um bit que é oposto ao estado de repouso antes do início da informação propriamente dita, e que é o bit de início e um ou mais bits de fim de byte (para que haja uma correcta separação entre os vários bytes) normalmente idênticos ao estado de repouso (figura 21).

Dado ser normal utilizar para estes bits a nomenclatura inglesa vamos passar a usá-la. Assim o bit de início é o ''Start bit'' e o(s) bit(s) de fim o(s) ''Stop bit''.

Dado ser possível a detecção de falsos ''Start bit'', para além de se utilizar normalmente uma técnica de reverificação deste bit (contrariamente ao processado com os outros), é também vulgar a utilização de um bit adicional por cada byte, o bit de paridade.

MODO SÍNCRONO

Se se garantir que existe sempre transmissão na linha mesmo quando não existe informação a transmitir torna-se supérfula a existência de bits de início e de fim, pelo que basta identificar determinados octetos bem definidos (os octetos cuja informação é nula e são enviadas precisamente quando não há informação a transmitir e que, por isso mesmo, têm um formato específico e bem definido—os bytes de sincronismo) e, a partir daí, todos os conjuntos de oito bits serão agrupados num byte.

Estes bytes serão enviados sempre que não haja informação a transmitir ou, quando essa informação for muito extensa, será intercalada por bytes de sincronismo para corrigir pequenos desajustes que eventualmente tenham surgido.

O facto deste modo de comunicação não utilizar os Start bit e Stop bit traz uma economia da ordem dos 20% no tempo de comunicação, pelo que o apontaria como modo preferencial da teleinformação. Contudo, a cadência de geração de informação (por exemplo, digitação de um teclado) é baixa, pelo que o tempo ganho na transmissão é dispendido no intervalo pelo que aquele valor se reduzirá muito. Acresce ainda que os equipamentos Síncronos, desde as saídas de computadores aos modems e terminais, são muito mais caros, que os equivalentes assíncronos. Talvez por isso é que a maior parte dos equipamentos com que os computadores comunicam são assíncronos. Veja-se que grande parte dos computadores compatíveis com os IBM PC trazem já de base uma porta de comunicação assíncrona, enquanto que não conheço nenhum que traga uma porta síncrona de base. Por isso, embora pretenda voltar a falar mais tarde das comunicações sincronas (nomeadamente os protocolos BSC, HDLC e SDLC) vamos, para já, apenas fazer uma análise mais detalhada às comunicações assíncronas.

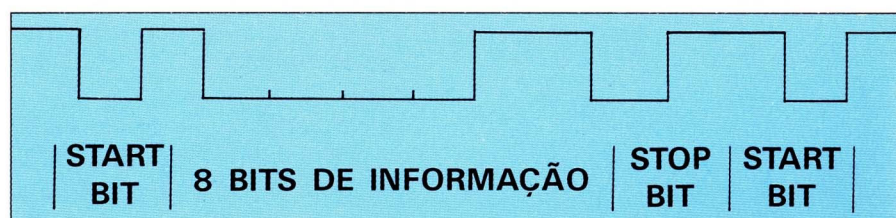
Foi já realçado que é necessário que o relógio de emissão e de recepção sejam o mesmo ou iguais.

Tão simples quanto isto!....

Para que o relógio seja o mesmo, basta determinar qual o equipamento que fornecerá esse relógio (cujo pino é por Standard o 24 da ficha de 25 pinos) e ligá-lo às duas entradas de relógio do mesmo Standard (15 e 17).

Só que... isto implica mais um condutor e, se houver modems intercalados,esta coincidência de relógio existirá apenas entre um modem e o equipamento a ele associado.

Mesmo que admitíssemos que era possível que dois sinais de relógio gerados em dois locais diferentes fossem absolutamente iguais, era necessário, além disso, garantir que estavam em fase, o que seria, pelo menos, complicado, mas que, a não ser garantido, falsearia a informação como a seguir se pode ver:

—Um sinal de relógio é uma onda periódica normalmente (mas não absolutamente necessário) quadrada, (ou seja, está tanto tempo no estado alto como no estado baixo), como aliás, foi já visto nos vários exemplos

atrás apresentados. Independentemente da duração ou, até certo ponto, da forma—que é normal ser quadrada, mas não necessariamente—, há um ponto único no sinal de relógio que determina a validação da informação (pois é essa a sua função), e que é a transição entre dois estados e, normalmente, só uma das transições.

Na figura 22 vemos um relógio de emissão e um relógio de recepção iguais mas desfasados de 180 graus. Em ambos os sistemas admitimos que o ponto de validade da informação é a transição do estado baixo para o estado alto. Na emissão não há lugar a qualquer dúvida, pois essa transição coincide com o meio do bit. Já na recepção a transição do relógio coincide com a transição da informação o que provoca de imediato uma indecisão na leitura.

Vejamos a figura 22b em que a transição da informação está representada de uma forma mais real (não existe mudança de estado instantânea). Considerando que o tempo de transição é T1, situando exactamente a meio do intervalo ΔT (muito curto), uma pequena variação provoca uma leitura errada. E faço notar que, nem sequer foi preciso admitir que os relógios não eram exactamente iguais.

Como resolver então o problema?

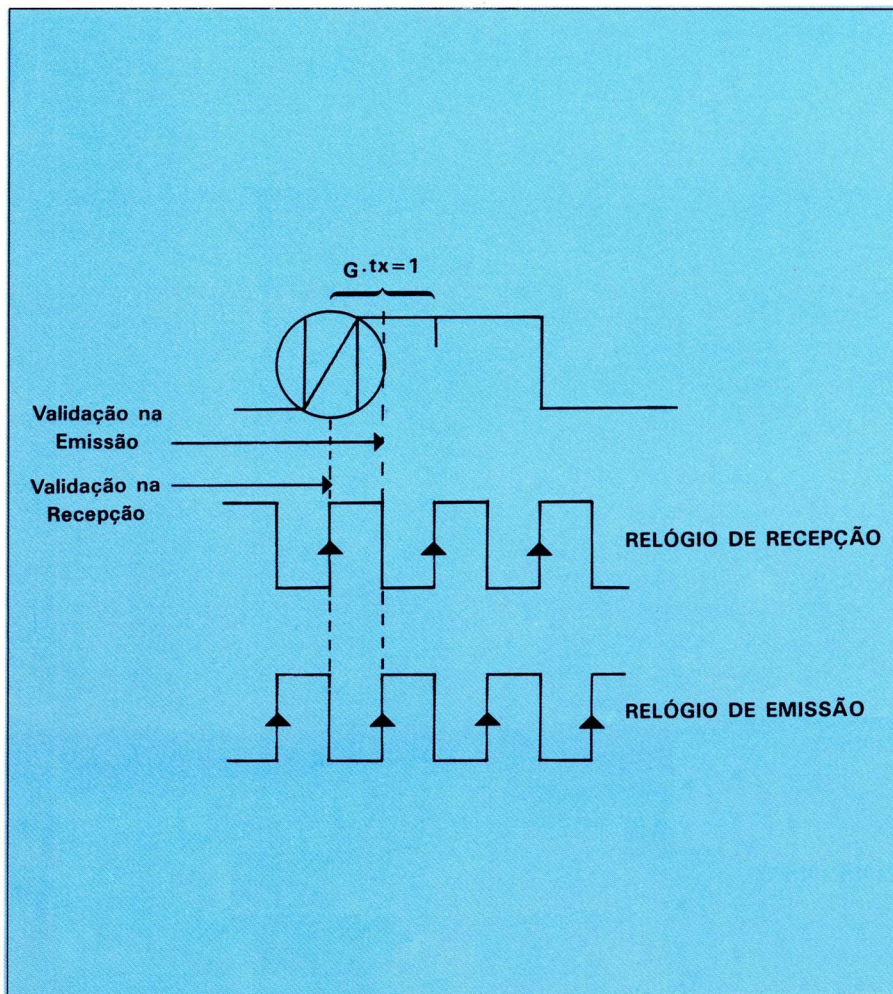
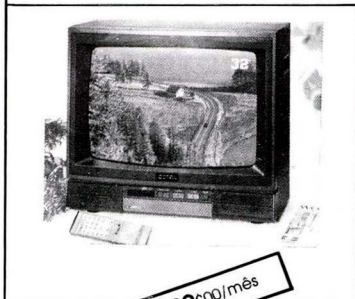
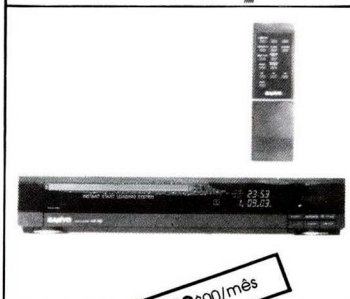
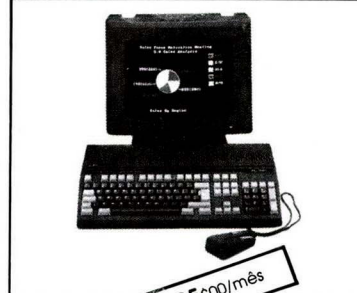
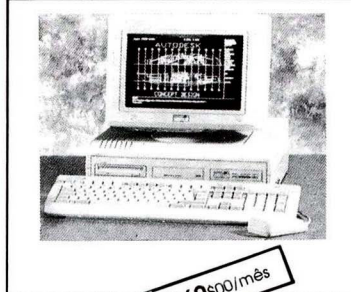
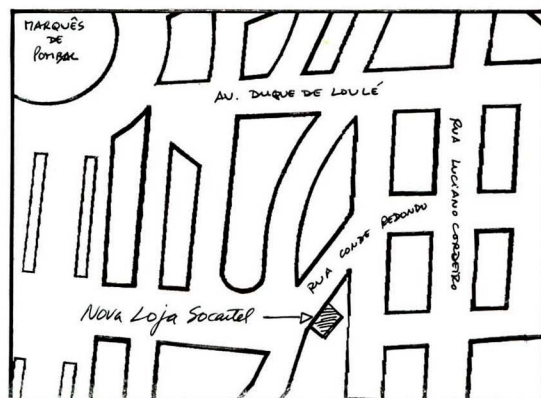
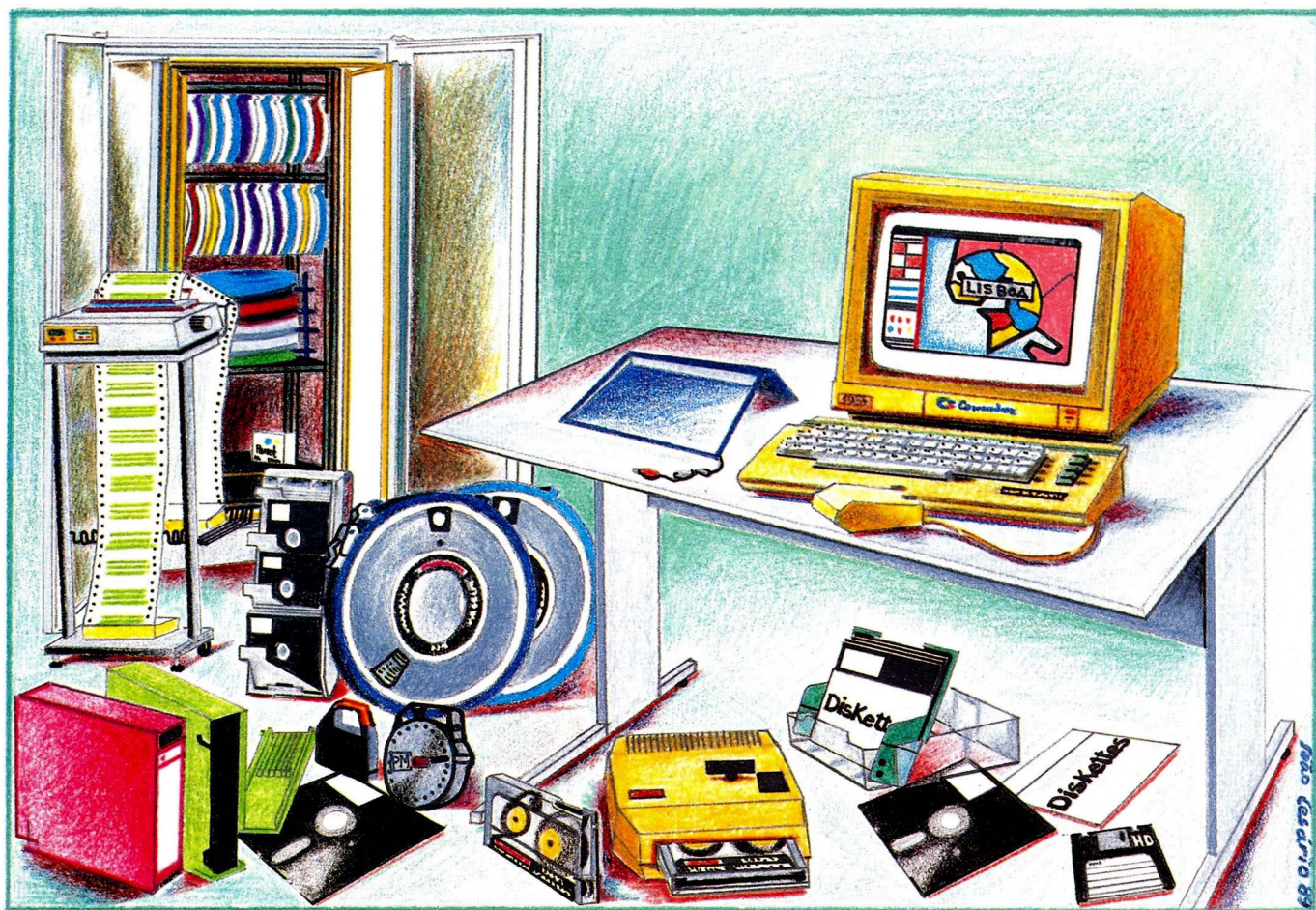
Simples. Nada obriga a que o relógio tenha a frequência da cadência do débito de informação, ou seja que se emita (ou receba) um bit em cada ciclo do relógio. Pelo contrário, se tiver um relógio cuja frequência é múltipla do débito de informação, basta garantir que se utiliza apenas o ciclo que mais se aproxime do meio do primeiro bit para validar a informação e todos os múltiplos deste. Qualquer variação razoável quer de frequência quer de fase entre emissão e recepção não traz qualquer erro de leitura, fig. 23.

Mesmo que haja um erro cumulativo razoável, este é perfeitamente recuperável pela existência de bits de sincronização (o bit de início e o(s) bit(s) de fim). É vulgar utilizar relógios que são 16, 32 ou 64 vezes superiores à cadência de informação.

E como é que se consegue que a transição válida se encontre mais ou menos a meio do bit a confirmar? Bem, na emissão parece-me evidente—até mesmo irrelevante... Na recepção, todos os ciclos do relógio provocam uma leitura da linha, (que estará em repouso, normalmente).

A partir do momento em que haja uma variação na linha, significará um Start bit. Então o circuito de recepção conta metade do factor da frequência de ciclos (ou seja se a frequência for 32 vezes a cadência, conta 16 ciclos, se for 16 vezes, conta oito...), confirma a permanência dessa variação (note-se que se consegue assim uma verificação do Start bit de que já atrás se havia falado, eliminando os falsos Starts bits), e, a partir daí, é só contar um número de bits igual ao factor e ir validando todos os bits até ao(s) Stop bit(s). A partir daí, todos os ciclos de relógio provocarão uma leitura da linha à procura de um novo Start bit.

O débito de informação é medido em bits por segundo ou em baud (em homenagem ao francês Baudot, pioneiro das comunicações). Embora frequentemente se confunda bits por segundo (b.p.s.) com baud, nem sempre estas duas grandezas são equivalentes, porque o baud refere-se ao número máximo de estados por segundo que um sinal pode ter e, recorrendo a determinados métodos de codificação (principalmente envolvendo MODEMS), é possível enviar mais que um bit por cada transição.

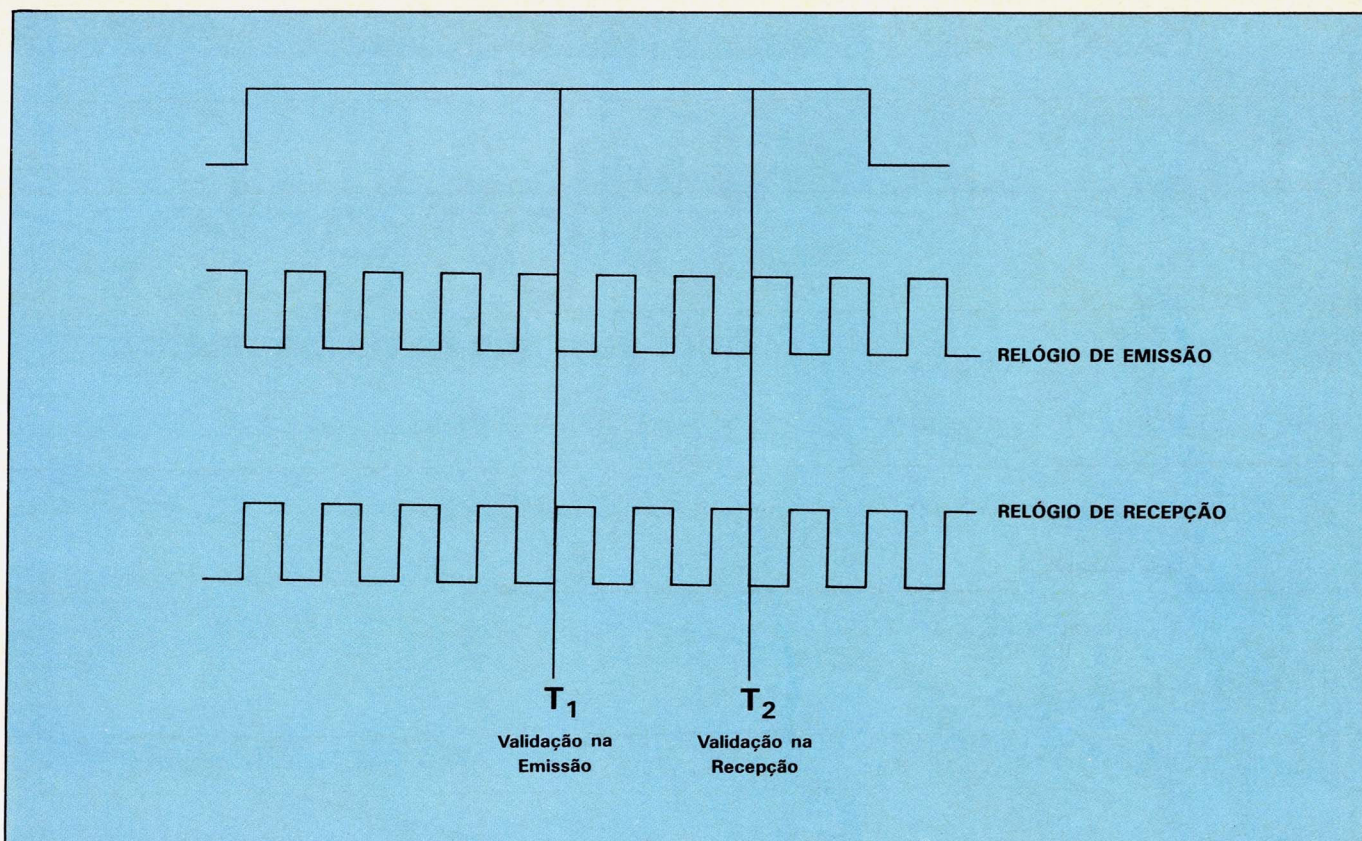
As velocidades de transmissão mais usuais são: 110, 150, 300, 600, 1200, 2400, 4800, 9600 e 19200.

O QUE DE BOM TEMOS PARA SI

Data Cartridges Streamer, Discos
Bandas Magnéticas, CALCULUS-EUROMAGNETICS
Diskettes
Fitas Tinta para Impressoras
Arquivo p/ Diskettes, Bandas, Discos, pastas, etc.
Suportes Rotativos p/ Terminais de Computadores
Monoblocos contra fogo p/ Registos Magnéticos
Etiquetas Autocolantes, Papel de Formulários
Diskettes de Limpeza p/ Unidades de Gravação
Pasta p/ Arquivo de Formulários e Conj. Separadores
Anti-Reflectores p/ Videos, Monocromáticos e a Cores
Mesas p/ Terminal e Impressora
Computadores COMMODORE
Impressoras STAR, C. ITOH
Máquinas de Limpeza de Óxidos p/ Cartridges (Streamers)
Visionador de Cabeças de Drive de Cartridges (Streamers)
Conjuntos de Limpeza p/ Drives de Diskettes, Cartridges, Teclados, Écrans, Impressoras

COMERCIALIZAÇÃO DE SUPORTES MAGNÉTICOS, LDA.

Sede:
Rua Artilharia Um, 39 - 1.º
☎ 69 34 37 - 69 34 08 Telex 64179
1200 LISBOA

DISTRIBUIDOR AUTORIZADO
COMMODORE E JOGOS

Filial:
Rua Damasceno Monteiro, 116 - B
☎ 82 01 85 - 82 77 36
1100 LISBOA

As velocidades mais comuns em comunicações com modems são 300 e 1200. Começam, contudo, a aparecer com uma certa frequência comunicações via Modem a 2400 (e a 1200,75—emissão a 75 baud e recepção a 1200, pois nada obriga a que a velocidade de emissão e recepção sejam iguais num determinado equipamento, desde que se passe o oposto no aparelho com ele conectado—recepção a 75 baud e emissão a 1200!—principalmente com a divulgação crescente do vídeo texto!).

As velocidades mais vulgares para a conexão de terminais são 9600 e 19200 com esta última a conquistar a hegemonia.

Embora a comunicação série hoje seja, em grande parte, utilizada para estabelecer a ligação entre um computador e os seus terminais, também é utilizada para ligação entre computadores e para débito de informação para uma impressora.

Em qualquer destes casos, muito especialmente no último, pode acontecer que haja necessidade de bloquear a comunicação, ou porque o terminal está desligado, ou porque o computador não pode guardar o ficheiro que o outro lhe quer enviar, ou ainda porque—o que é vulgar—a velocidade de escrita da impressora é mais lenta que a velocidade de transmissão dos caracteres a imprimir, ou mesmo porque está sem papel, ou ainda ''off line'', pelo que é necessário implementar um controlo de fluxo. Este controlo de fluxo pode implementar-se recorrendo a sinais específicos reservados, já para esse efeito, pela norma RS232, e que são o DSR, DTR e RTS, CTS *como será descritivo no próximo número*, ou pela utilização de um protocolo de Software.

Muitos protocolos foram estabelecidos para gerir o controlo do fluxo de uma comunicação. Se a comunicação se efectuar entre dois computadores qualquer pessoa pode facilmente projectar e implementar um protocolo de comunicação que, acredito, seja mais eficaz que os protocolos já existentes, dado poder contemplar a situação específica a que se destina.

De todos eles, o mais conhecido é o XON, XOFF. É um protocolo muito simples. O receptor, qualquer que ele, seja envia ao emissor o código 19 (13H) sempre que pretenda que este suspenda a emissão e o código 17 (11H) para que a retome.

Na prática, o receptor tem uma memória tampão, manda suspender a comunicação sempre que essa memória se encontre quase cheia e manda recomeça-la quando a memória está quase vazia.

O protocolo ETX/ACK é muito utilizado na transferência de ficheiros. É também muito simples. A informação é enviada em blocos de comprimento fixo. O emissor envia sempre um ETX (código 3—fim de texto) para informar o receptor que terminou o bloco. Só então o receptor responderá com o caracter ACK (código 6, recepção correcta)—ou o caracter NACK (código 21—recepção com erro) e, neste caso, o emissor repetirá outra vez o último bloco.

O protocolo RS232 foi definido essencialmente para gerir a conexão entre um computador e um MODEM, pelo que este se rege sempre pelo controlo de fluxos dos sinais hardware. Os protocolos de comunicação foram desenvolvidos principalmente para gerir as transferências entre computadores. Assim, se se fizer uma transferência entre computadores, via MODEM, temos uma combinação de controlos de fluxo, por hardware e por software.

ARQUIVOS
DE DISKETTES
boeder
... onde quer que exista informática
ARQUIVOS
P/ 10 DISKETTES
CAIXA ARQUIVO
EXPANDÍVEL
KIT DE DISKETTES
DE LIMPEZA
TOALHETES
DE LIMPEZA
KIT DE LIMPEZA
MARGARIDAS E ESFERAS
BASE
P/ MOUSE
ETIQUETAS
P/ DISKETTES
DISKETTES BOEDER
(garantia alemã de 5 anos)
conheça a mais completa linha
de produtos para informática
IMPORME
Soc. Importadora de Material de Escritório, Lda.
Praceta Pedro Escobar, 40-r/c • Loja A
Tels. 308472 / 307325 • Fax 305794 • 4400 V. N. GAIA

DESTACÁVEL

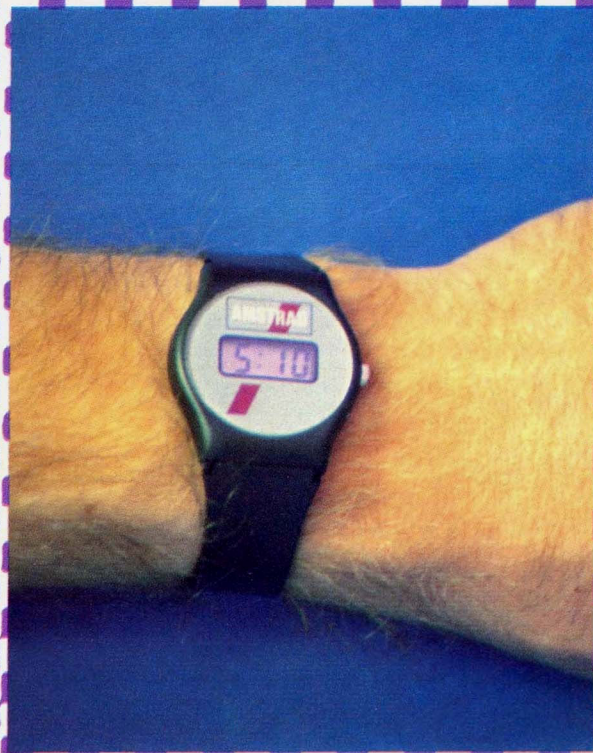
clube

AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

PROGRAMAS DISPONÍVEIS

VER DESCRIÇÃO NOS NÚMEROS ANTERIORES DA AMSTRAD MAGAZINE

- FS-101 BUGS
- FS-102 PINBALL
- FS-103 PITFALL
- FS-104 POKER MACHINE
- FS-105 PYRAMID
- FS-106 RAIN
- FS-107 ROCKETS
- FS-108 XWING
- FS-109 MAHJONG
- FS-110 MATH PAK
- FS-111 EPISTAT
- FS-112 MAHJONG — para ecrã EGA
- FS-113 ALLMAC
- FS-114 ICON MAKER
- FS-115 ALTAMIRA — editor gráfico
- FS-116 DRAW POKER
- FS-117 PIANO MAN
- FS-118 UTILITÁRIOS PARA ECRÃS EGA
- FS-119 WORLD
- FS-120 MUSIC
- FS-121 PAINT
- FS-122 FXMATRIX
- FS-123 BIORRITMO VERSÃO 3.0
- FS-124 TAROT
- FS-125 BLACK JACK
- FS-126 GIN RUMMY
- FS-127 EDWIN
- FS-128 MONOPOLY
- FS-129 ANSIDRAW
- FS-130 CASIOZ
- FS-131 BIORRITMO PESSOAL
- FS-132 BACCARAT
- FS-133 I`CHING
- FS-134 ANSI-ANIMATOR
- FS-135 MAIL
- FS-136 LABEL
- FS-137 TEMAS MUSICAIS
- FS-138 TWCALC22
- FD-904 ORIGAMI
- FS-140 GAMÃO
- FS-141 PRODIAGS
- FS-142 EMULADOR DE Z80 E CP/M 2.2
- FS-143 SPOOLER P/ MS-DOS
- FS-144 EMULADOR DE CGA PARA CARTA GRÁFICA HERCULES
- FD-901 STAR-SAK, PC-SIZE, FORGET-IT, PC-PLAN, PC-EMS, PC-MULTI, PC-PITMAN
- FD-902 TRIVIA MACHINE
- FD-903 UTILITÁRIOS PARA O WORDSTAR
- FS-145 FRED
- FS-146 BAS-INIT
- FS-147 YAHTZEE
- FS-148 DGEDIT-EDI. PESSOAL
- FS-149 L5
- FS-150 SHORTIES
- FS-151 DROP
- FS-152 GEMDOS. BAT
- FS-153 FILE SCAN V2.0

aprenda você mesmo...

D BASE III PLUS - LOTUS 123 — PC/MS-DOS

A QUALQUER HORA, EM QUALQUER LUGAR... FÁCIL PARA TODOS

Divididos em vários módulos, os cursos APRENDA VOCÊ MESMO tornam-se muito flexíveis.

Acabam-se os horários rígidos. Os alunos estudam quando têm tempo, repetindo as lições até se familiarizarem com o tema.

A aprendizagem faz-se ao ritmo individual de cada um e pode fazer-se em qualquer sítio onde exista um computador.

Funcionando com os computadores mais populares, os cursos podem ser partilhados por várias pessoas que, assim, estabelecem o horário que mais lhes convém.

Mas uma das grandes vantagens dos cursos APRENDA VOCÊ MESMO, é que foram especialmente concebidos para se adaptarem aos diferentes níveis de conhecimento dos formandos.

Como indica a própria designação APRENDA VOCÊ MESMO, os cursos são facilmente utilizáveis pelo próprio formando, sem necessidade de um instrutor.

Basta introduzir uma disquete no computador e seguir as instruções visíveis no écran. Isto torna os cursos altamente acessíveis, mesmo para as pessoas sem quaisquer conhecimentos de informática.

Além disso, todo o conteúdo é apresentado em português não existindo, portanto, as barreiras linguísticas habituais nas novas tecnologias.

CONTEÚDO DOS CURSOS

- PC/MS-DOS
  Estrutura de um computador
  — Formatar/verificar uma disquete
  — Visualizar o conteúdo
  — Apagar/copiar ficheiros
  — Copiar a disquete inteira
  — As teclas de função
  — Ficheiros Batch
  — Directorias
  — Discos virtuais.

REF. 400

- LOTUS 1 - 2 - 3
  Noções de base
  — Movimentos
  — Números e textos
  — Fórmulas e cálculos
  — Copiar
  — Guardar o trabalho
  — Janelas
  — Edição dos dados
  — Gráficos
  — Base de dados
  — Macros.

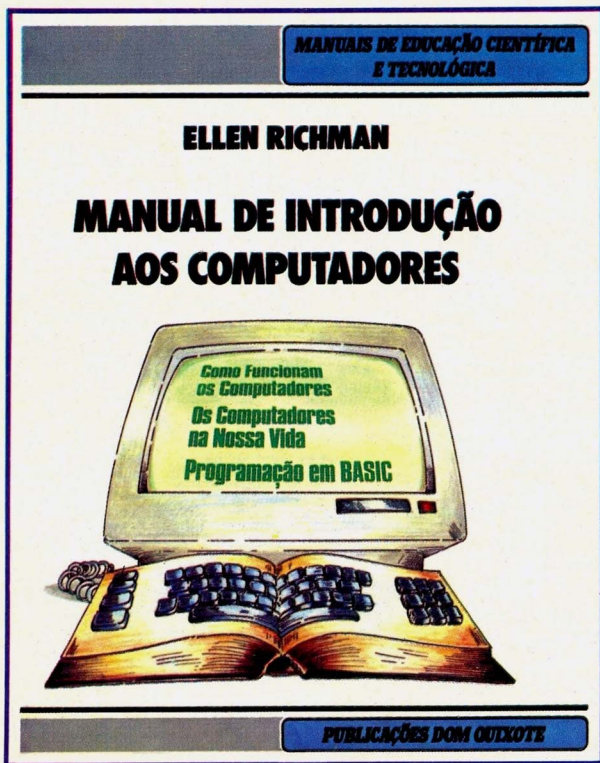
REF. 401

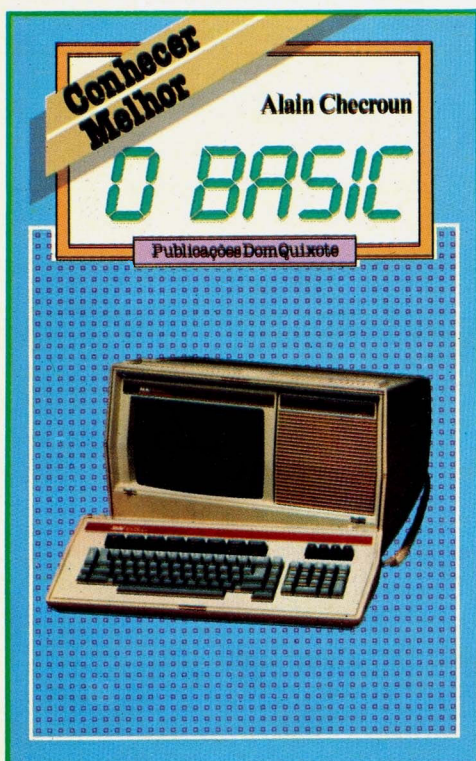
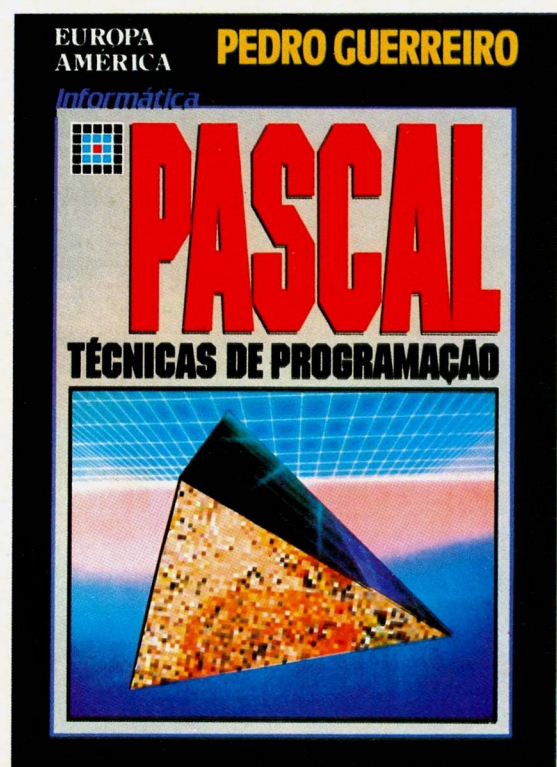
- dBASE III PLUS
  Noções sobre a base de dados
  — Introdução/Visualização/Criação/Modificação
  — Busca de registos
  — Índice
  — Impressão
  — A gestão de menus
  — O cálculo
  — A programação
  — Introdução de dados
  — Leitura de dados
  — Ordenação

REF. 402

APRENDA VOCÊ MESMO	PC/MS - DOS	REF. 400	19.900$00
APRENDA VOCÊ MESMO	LOTUS 1-2-3	REF. 401	19.900$00
APRENDA VOCÊ MESMO	dBASE III +	REF. 402	19.900$00

(Não se esqueça de indicar o formato das disquetes

USE **POSTAL N.º 4**

REF. 910 1 400$00

PREÇO: 1 450$00 REF. 913, postal 3

REF. 911 880$00

PREÇO: 2 480$00 REF. 912, postal 3

• VER AS DESCRIÇÕES NOS NÚMEROS ANTERIORES

NOVIDADE

A informatização da sociedade foi sem dúvida o facto mais marcante do último decénio e ninguém pode, hoje em dia, ignorar o peso crescente da informática nos mais diversos campos da actividade quotidiana.

É neste contexto que o presente dicionário visa um duplo propósito:

- ao leitor comum ele explica, de um modo simples e acessível, o sentido dos principais termos utilizados, ao mesmo tempo que, através de artigos mais desenvolvidos sobre os conceitos-chave, permite compreender as grandes linhas da informática actual;
- ao especialista fornece um panorama completo sobre a evolução da técnica e da terminologia informática, facultando-lhe simultaneamente as correspondências, em inglês e francês, para cada uma das palavras ou expressões já consagradas na língua portuguesa.

PREÇO: 1 800$00 postal 3 REF. 915

Será que os computadores só podem ser utilizados por quem sabe inglês?
É evidente que não. Embora o conhecimento da língua inglesa facilite a aprendizagem, nunca se poderá considerar indispensável para este efeito. No nosso país, são cada vez mais frequentes as marcas que traduzem os manuais e as packages, e adaptam os teclados, para poderem possuir boas soluções informáticas em mercados que nada têm a ver com a língua inglesa.
Foi assim, seguindo esse princípio, que AM optou por incluir nesta secção a tradução do MANUAL DO PC, para facilitar a vida a todos os que em Portugal preferem ler em português.

PREÇO: 1 900$00 REF.310, postal 3

Complete a sua colecção

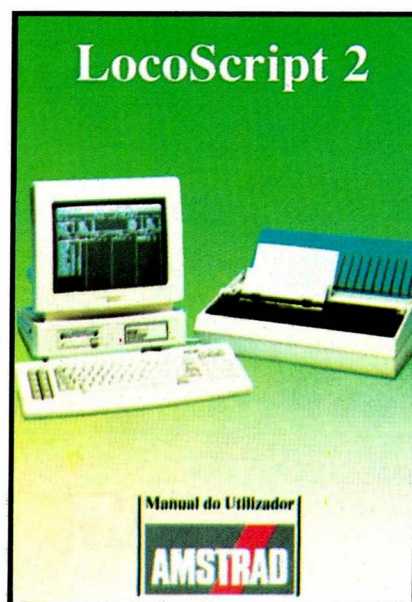
200$00 cada exemplar atrasado

USE **POSTAL N.º 6**

DMP 4000
- MANUAL DE UTILIZAÇÕES EM PORTUGUÊS

Com uma qualidade de impressão relativamente elevada tendo em consideração que se trata de uma impressora de 9 agulhas, a DMP 4000 pode distinguir-se actualmente como uma impressora bem sucedida no mercado nacional. Tal facto, constituiu uma das razões que nos levou a optar pela inclusão do seu manual de utilização, EM PORTUGUÊS, nesta secção da AM, procurando com isso continuar a proporcionar aos nossos leitores informação tão detalhada quanto possível, numa linguagem tão simples quanto possível, a um preço nitidamente impossível.

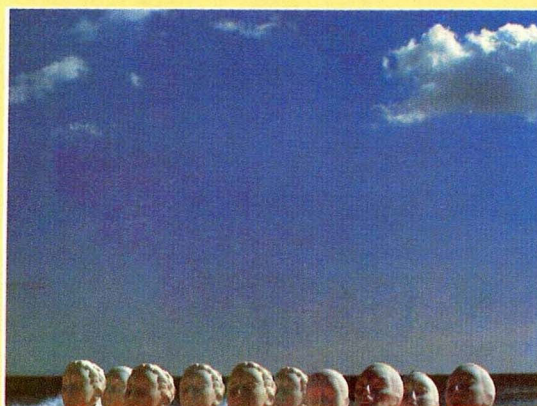
PREÇO 500$00 REF. 320, POSTAL 3

LOCOSCRIPT 2 (para PCW 9512) Manual do Utilizador EM PORTUGUÊS

Quase quatrocentas páginas de texto, figuras, esquemas, e exemplos, constituem o mais completo livro em português sobre um processador de texto que tem arrastado centenas de pessoas dos teclados das máquinas de escrever para os teclados das modernas máquinas de processamento de texto.

PREÇO: 1 200$00 REF. 322, postal 3

universidade moderna
82

Helder Coelho

TECNOLOGIAS DA INFORMAÇÃO

Publicações Dom Quixote

As tecnologias da informação (TI's) — ou os três C's, Computador, Controlo e Comunicação — dizem respeito às capacidades de solucionar os problemas e de responder às necessidades através do recurso à informação. As tecnologias do computador abordam a compreensão, processamento, armazenamento e pesquisa da informação. As tecnologias do controlo suportam o acompanhamento dos processos, das ferramentas e dos instrumentos. As tecnologias da comunicação cobrem a automatização do encaminhamento e comutação da informação.

Hoje em dia, nos países industrializados, estamos em presença de uma situação caracterizada por processos automatizáveis por computador e por produtos bastante mais complexos, pela introdução constante de novos artigos, e por mercados evolutivos. Portugal não pode alhear-se destas mutações tecnológicas e económicas, ficando reduzido a um mero espectador/utilizador.

Ao longo deste livro o autor dá uma panorâmica dos diversos esforços internacionais, do real significado destas tecnologias e das orientações associadas ao seu desenvolvimento, apresentando as suas ideias quanto à aplicação das TI's em Portugal no sentido de constituirem uma das bases técnicas do nosso desenvolvimento.

NOVIDADE

GESTÃO DE CONTA BANCÁRIA

A sua conta bancária actualizada dia a dia no seu próprio computador

PREÇO: 20 900$00 REF.413, postal 4

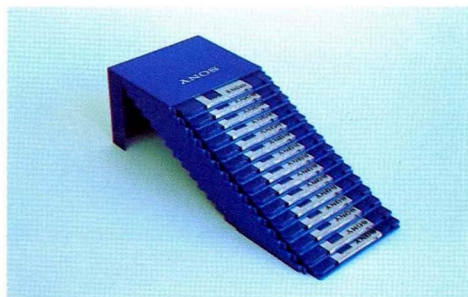
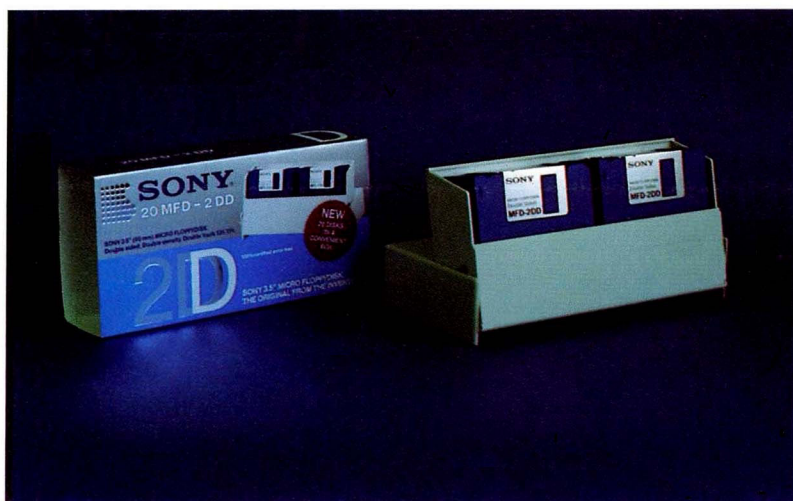
Não se esqueça de indicar o formato das disquetes

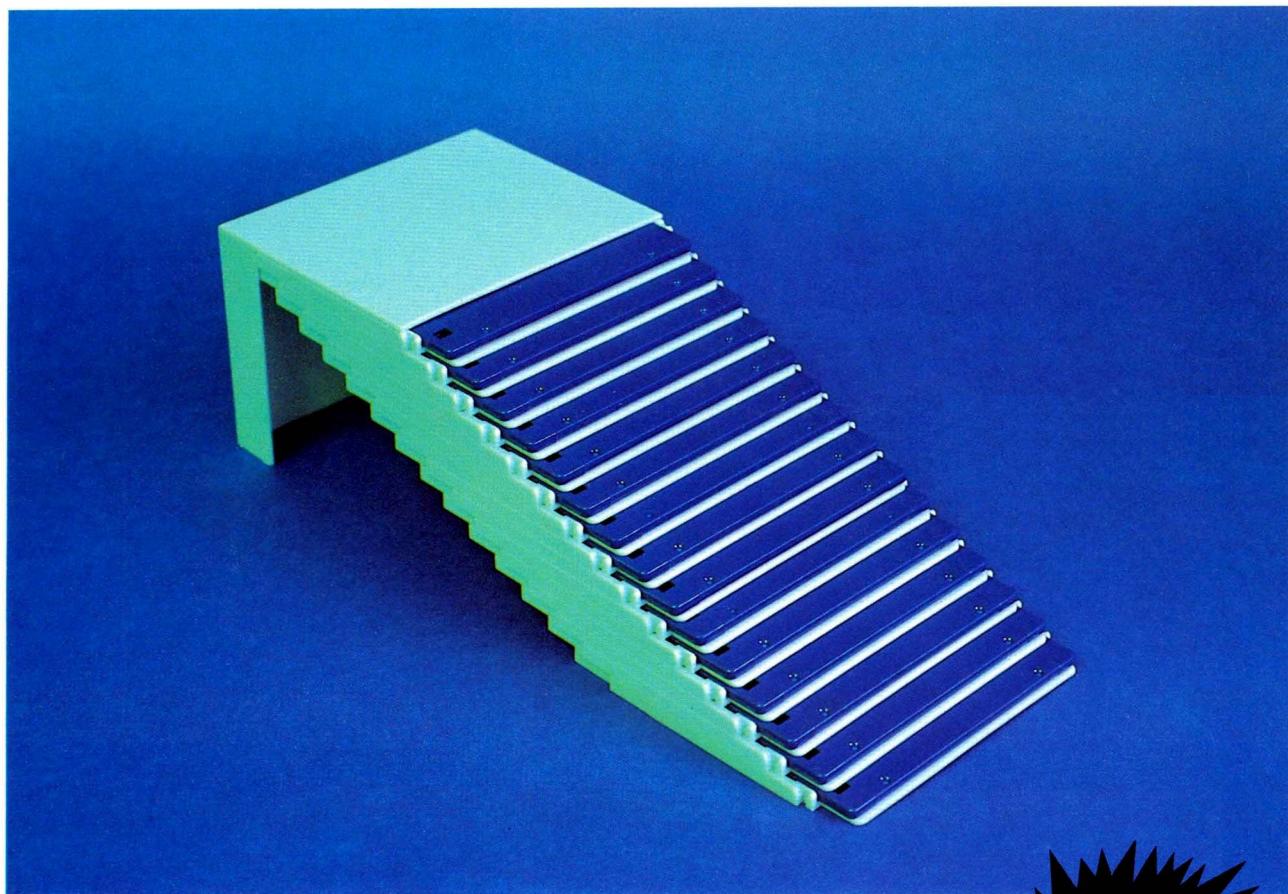
ARQUIVADOR DE 40 DISKETTES DE 3,5'' COM FECHO E SEPARADORES

REF. 605 — 3 490$00

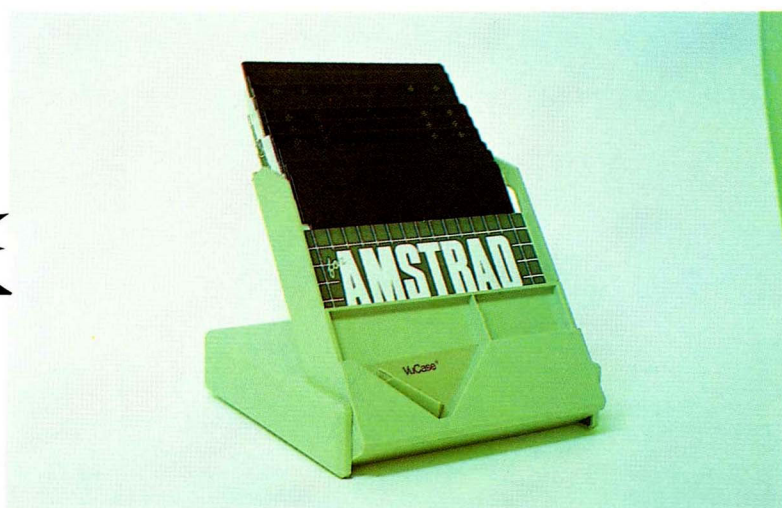
ARQUIVADOR DE 50 DISKETTES DE 5,25'' COM FECHO E SEPARADORES

REF. 603 — 3 190$00

CAIXA DE 10 DISKETTES SONY (3.5'' - 720K) COM OFERTA GRÁTIS DE FICHEIRO

REF. 700 — 5 990$00

CAIXA DE 20 DISKETTES SONY (3,5'' - 720K) COM FICHEIRO

REF. 701 — 11 490$00

NOVIDADE

FICHEIRO PARA DISKETTES DE 3,5''

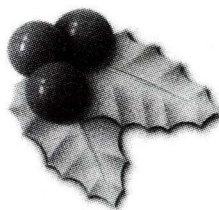
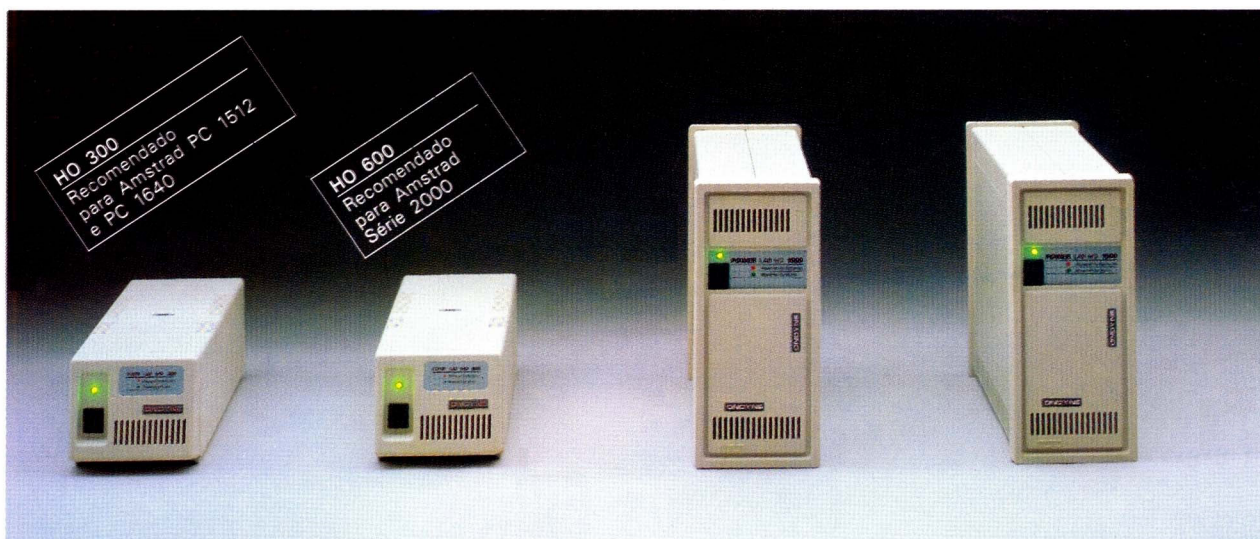
REF. 607 1 490$00

FICHEIRO PARA 10
DISKETTES DE 5,25''

É TÃO FÁCIL

REF. 608 — 2 190$00

FICHEIRO PARA 12
DISKETTES DE 3,5''

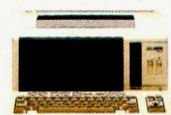
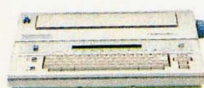
REF. 609 — 2 290$00

CARACTERÍSTICAS:
• Processador: 8086
• Memória: 512 K
• Velocidade: 8 MHZ
• Ecrã: CGA (mono ou cores)
• Um drive 720 K
Os Seus Filhos Vão Adorar!
Para os mais novos, o Natal é uma época muito especial. Os seus filhos também sentem isso. Este ano, Você vai poder oferecer-lhes uma prenda que eles nunca esquecerão: um SINCLAIR PC 200.
Um verdadeiro computador pessoal para eles utilizarem nos tempos livres ou nas tarefas escolares com a garantia de compatibilidade com os sistemas profissionais futuros. Consulte uma loja SOCARTEL e faça com que este Natal seja ainda mais feliz.
SOCARTEL
LOJAS SOCARTEL: LISBOA, OLHÃO, PORTIMÃO, SINES, CARTAXO, LEIRIA, PORTO, COIMBRA, MONÇÃO, GUIMARÃES, PÓVOA DE VARZIM, CHAVES, BRAGANÇA, PENAFIEL

ONDYNE

SISTEMAS DE ALIMENTAÇÃO ININTERRUPTA PARA MAIOR SEGURANÇA NO SEU TRABALHO

CARACTERISTICAS TECNICAS

GAMAS		MODELOS	MODO DE FUNC.: ON-LINE	MODO DE FUNC.: OFF-LINE	CARACTERISTICAS ELECTRICAS – ENTRADA: MONOFASICA	ENTRADA: TRIFASICA	ENTRADA: TENCÃO 220 V	ENTRADA: TENCÃO 380 V	ENTRADA: FREQUENCIA 50 HZ.	SAIDA: MONOFASICA	SAIDA: TRIFASICA	SAIDA: TENCÃO 220 ± 10%	SAIDA: TENCÃO 380 ± 5%	SAIDA: FREQUENCIA 50 HZ.	POTENCIA VA - KVA
COMUTAÇAO INFERIOR A 1 ms.	HØ	HØ 300	N	S	S	N	S(1)	N	S(4)	S	N	S	N	S(6)	300
		HØ 600	N	S	S	N	S(1)	N	S(4)	S	N	S	N	S(6)	600
		HØ 1000	N	S	S	N	S(1)	N	S(4)	S	N	S	N	S(6)	1.000
		HØ 1500	N	S	S	N	S(1)	N	S(4)	S	N	S	N	S(6)	1.500

BATERIAS: P-B-CA S/MANUTENÇAO RECARGA TIPICA 10 K.	BATERIAS: TIPO (AMP/H) 6.5	TIPO (AMP/H) 24	TIPO (AMP/H) 38	ONDULADOR: ONDA DE SALDA SINUSOIDAL	ONDULADOR: TECNO. FILTRO	TECNO. P.W.M.	BY-PASS	TEEPO COMUTA. < 1 msg.	TEEPO COMUTA. 0 msg.	CARACTERISTICAS FISICAS: MEDIDAS COMPRIMENTO	MEDIDAS LARGURA	MEDIDAS ALTURA	PESO (Kg.)
S	S	N	N	S	S	N	N	S	N	365	165	140	11
S	S	N	N	S	S	N	N	S	N	365	165	140	15
S	S	N	N	S	S	N	N	S	N	360	158	436	30
S	S	N	N	S	S	N	N	S	N	360	158	436	40

ONDYNE HO 300	REF. 610	PREÇO: 99 900$00
ONDYNE HO 600	REF. 611	PREÇO: 127 900$00
ONDYNE HO 1000	REF. 612	PREÇO: 249 900$00
ONDYNE HO 1500	REF. 613	PREÇO: 324 900$00

MICROSOFT FLIGHT SIMULATOR version 3.0

Suporta todas as cartas gráficas desde CGA a VGA

Para quem gosta de simuladores de vôo este é **O SIMULADOR DE VÔO**.

Suportando muitas das cartas gráficas habituais nos PC´s inclusivé a Hercules, a EGA, a VGA, e a CGA em visores de cristal liquido ou CRT´s, o Flight Simulator que neste número colocamos à disposição de todos os leitores foi concebido por uma das maiores softhouses da actualidade, senão mesmo a maior - a Microsoft - e é no mínimo um simulador excelente a todos os níveis. Em termos de gráficos, por exemplo, para além de suportar as cartas gáficas já referidas e de delas extraír as capacidades que lhes são inactas, suporta ainda outras cartas gráficas não previstas na versão base mas adicionaveis através de drivers externos.

A simulação que pode decorrer num de três aviões diferentes, escolhido pelo utilizador, pode basear-se em operações de descolagem, aterragem, ou vôo normal, sofrendo, ou não, efeitos climatéricos (chuva, vento, neve, etc), ou temporais (dia, fim de tarde, noite, etc.), e estando, ou não, condicionada a um conjunto enorme de outros factores, entre os quais podemos referir os vôos em esquadrilha, ou em perseguição, quer em periodos de paz, quer em periodos de guerra.

O nível de realidade da simulação e controlavel pelo utilizador através de opção acedida por teclado, e para os utilizadores menos à vontade num "cock pit" existe ainda a possibilidade de assistir a lições de vôo sub-divididas por tarefas a executar. A documentação é composta por um enorme manual, diversos mapas, e um pequeno livro de "Quick Reference" (referências rápidas), apoiando de uma forma melhor do que excelente o jogo que se encontra dividido pelas duas disquetes de 5.25" que complementam a package.

Para além do interesse do jogo, pensamos que é digno de nota o facto dele suportar e tirar proveito das cartas VGA, facto que, sem dúvida, o torna único no mercado português.

PREÇO: 9 900$00 **REF. 330, postal 3**

QUICK BASIC versão 3.0

SUPORTA O PROCESSADOR ARITMÉTICO 8087

Uma excelente linguagem de programação e tum óptimo compilador de programas concebidos em BASICA ou GW-BASIC, o Quick BASIC proporciona a todos os programadores desta linguagem uma velocidade de processamento que embora não sendo tão grande como a que se obtém no dialecto da mesma linguagem lançado pela Borland, é muito mais standard.

Para todos os utilizadores do GW, o Quick BASIC só pode ser a evolução perfeita. Baseado num set de instruções que quase se pode considerar cem por cento igual ao do dialecto GW, o QB traz-nos toda a velocidade de uma linguagem compilada, as facilidades de "debugging" comuns aos interpretadores da mesma linguagem, e um completo manual de utilização, por um preço impossivelmente baixo!!!

PREÇO: 19 900$00 REF. 331, postal 3

a-quatro

EDIÇÃO TEXTO
- Formato A4/Formato texto
- Modo linha/Modo coluna
- Manipulação blocos — Cópia — Movimentação — Remoção

PROCESSAMENTO TEXTO
- Régua de «Tabulação»
- Definição de margens — Esquerda — Direita
- Alinhamento margens definidas (Word-wrap)
- Reformatação
- Pesquisa/Substituição de texto

IMPRESSÃO TEXTO
- Definição comprimento página impressora
- Documento activo/Documento previamente gravado
- Composição efeitos especiais de impressão

contabilidade

- Aplicação dirigida a utilizadores com reduzidas operações contabilísticas.
- Introdução de lançamentos funcional.
- Numeração automática dos documentos.
- Descrição completa de movimentos.
- Lançamento por diário e documento.
- Controlo automático do valor por documento e por diário.
- Aceitação provisória de documentos em erro, permitindo a sua correcção ou conclusão posterior.
- Extractos de conta.
- Balancetes de razão, balancetes de contas gerais e balancetes analíticos com valores mensais ou acumulados de qualquer mês.
- Consulta do estado de qualquer conta no ecran, por mês.
- Possibilidade de abertura de novo exercício sem encerrar o anterior.

A QUATRO	REF. 411	8 900$00	Postal 4
CONTABILIDADE	REF. 412	8 900$00	Postal 4

(Não se esqueça de indicar o formato das disquetes)

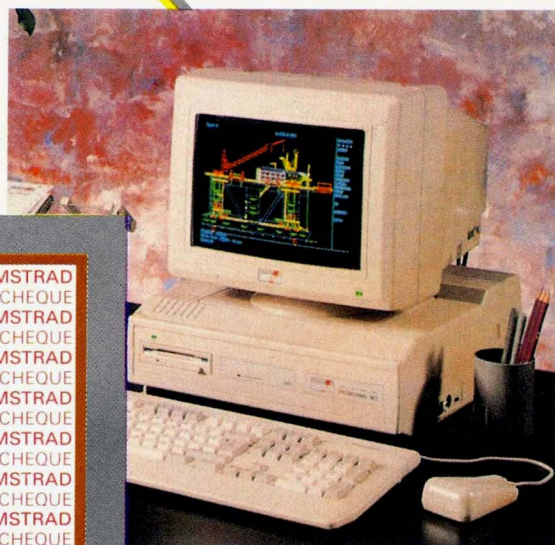
O F E R T A E

SOCARTEL

VALE 5 000.00

na compra
a pronto
de uma aparelhagem
AMSTRAD
MCD 7

VALE 10 000.00

na compra
a pronto
de um
AMSTRAD
PC 2086

S P E C I A L

AMSTRAD MAGAZINE

REVISTA DOS UTILIZADORES AMSTRAD

CHEQUE AMSTRAD

VALE 7 500.00

na compra a pronto de um SINCLAIR PC 200

L O J A S

SOCARTEL

LISBOA
- Av. Eng. Duarte Pacheco, 17
- R. Conde Redondo, 127-A

OLHÃO
- Rua João Rosa, 6-A

PORTIMÃO
- Bairro do Pontal - Bloco 2-A, c/v

SINES
- Av. General Humberto Delgado, nº 43

CARTAXO
- Centro Comercial A. C. Santos - Loja 4

LEIRIA
- Quinta Sto. Antônio, 67 - Loja 1

PORTO
- Av. Boavista, 2881 - Loja 3
- Rua Sta. Catarina, 716

COIMBRA
- Centro Comercial Primavera
  Av. Calouste Gulbenkian, Lote 7 - Loja 37

MONÇÃO
- Rua da Independência, 16, r/c

GUIMARÃES
- R. Manuel Saraiva Brandão, 241, r/c

PÓVOA DO VARZIM
- Praça dos Combatentes - Loja 18

CHAVES
- Edifício Europa - Av. 5 de Outubro - Loja 1

BRAGANÇA
- Edifício Translande
  Av. Sá Carneiro, Bloco 1 - Loja 19

PENAFIEL
- Edifício Brasília - Loja X

SETÚBAL
- Travessa do Carmo, 23

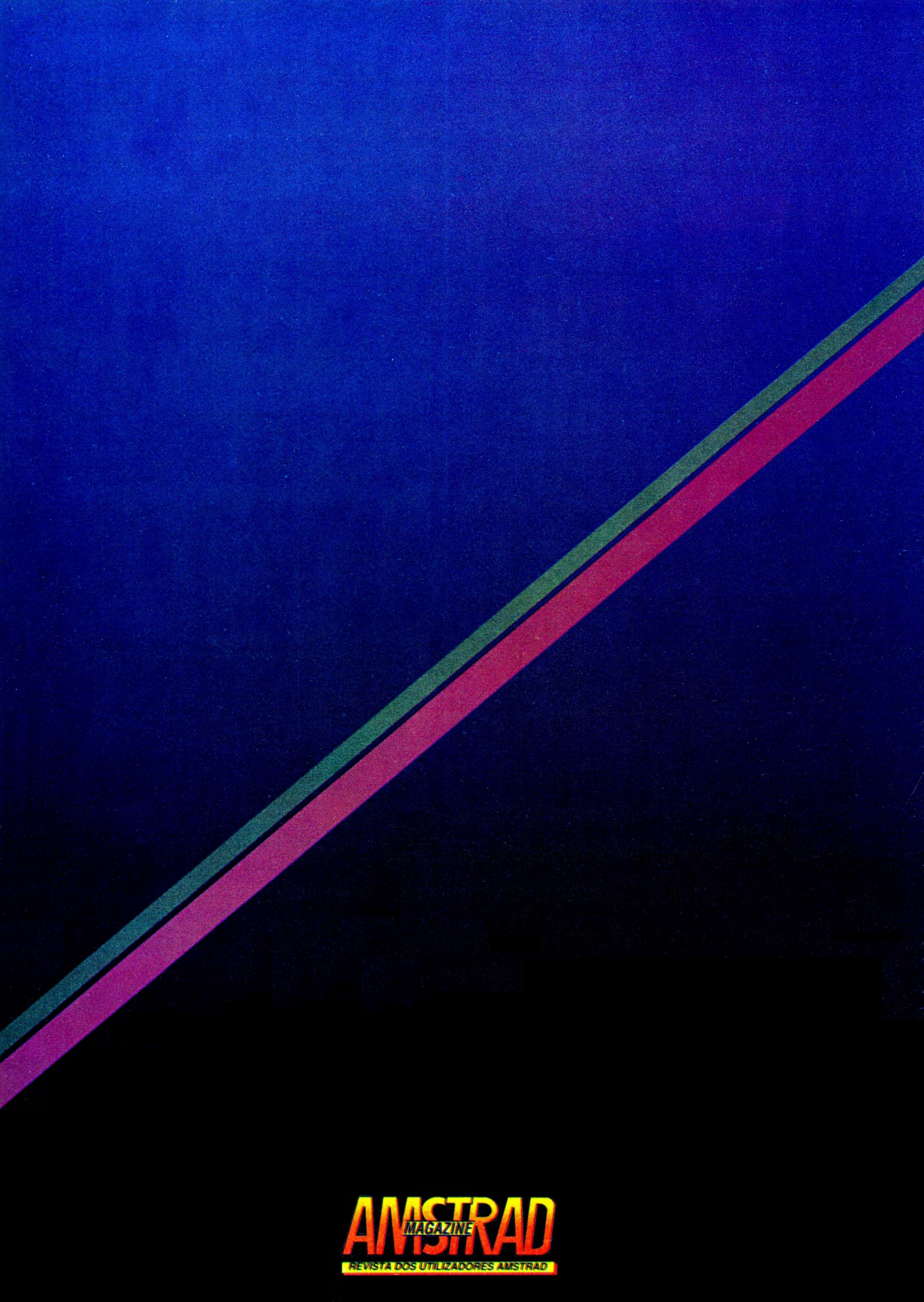
FAMALICÃO
- R. Adriano Pinto Basto, 134

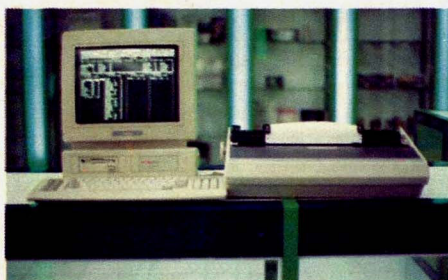
VILA DO CONDE
- R. Dr. Pereira Júnior, 53 - Loja 4

VILA REAL
- Centro Comercial Milagorgo - Loja 13

FIGUEIRA DA FOZ
- R. Dr. Francisco Antônio Dinis
Complexo Mondego - Loja 14

AVEIRO
R. Dr. Alberto de Souto
Ria Plano - Loja 27

SAMORA CORREIA
Centro Comercial Samora Correia
Urb. Arneiro dos Corvos, 23

UM POUCO DE HISTÓRIA

A caminho da perfeição

Os computadores Amstrad detêm a melhor relação qualidade/preço. Aqui contamos, muito sucintamente, o caminho da micro-informática dos últimos anos.

É normal que um potencial comprador de PC compatível se sinta desorientado face à variedade de configurações que lhe são propostas. Aparecidos em 1983, os PC não pararam de evoluir tecnicamente. Se você apanhou o comboio já em marcha, é bom que possua os conhecimentos necessários para distinguir o milho do joio.

Até ao aparecimento dos PC's compatíveis, a micro-informática era uma autêntica anarquia: quase trinta construtores lançavam máquinas completamente incompatíveis entre si. Utilizavam os microprocessadores de oito bits (6502 e Z80) onde, com uma RAM máxima de 64K, completamente insuficiente para os profissionais que pretendiam libertar os seus pesados computadores das pequenas aplicações locais. Destacaram-se três máquinas; Apple IIe, Commodore 64 e, depois, l'Oric Atmos. Os micros eram pobres em performances, havia necessidade de um standad.

A IBM ENTRA EM CAMPO

O gigante IBM começa então a interessar-se pela micro informática e lança o seu PC (computador pessoal), acordando com a Microsoft o desenvolvimento do software respectivo. Para a época foi uma super-bomba técnica! Senão, vejamos só:

- —microprocessador 8088 podendo endereçar até 20 bits.
- —frequência de relógio de 4,77 MHertz.
- —leitor de disquetes de 360 k.
- —sistema operativo MS-DOS super prático (é o CP/M da Digital Research muito melhorado).
- —um Basic quatro vezes mais rico que o da Apple.
- —96K de Ram extensível até 640K.

—teclado com dez teclas de funções e secção numérica.

—tudo é modular, intermutável, tanto em hardware como em software. É uma solução cara mas segura para o comprador.

DO PC AO PS

Em 1984 a Apple lança o seu primeiro Macintosh. Era um autêntico ''carro de bois'' e ainda mais caro que o IBM. A IBM abre então o jogo autorizando os outros fabricantes a fazer compatíveis IBM PC assim como placas para estas máquinas. A concorrência obriga os preços a baixarem. Durante estes tempos a Apple melhora o seu Mac mas fecha-se no seu célebre proteccionismo absoluto. A micro-informática profissional torna-se 70% PC e 20% Mac. O PC torna-se, de facto o standard apesar dos seus defeitos.

Mais tarde a IBM compreende que a sua criança lhe escapa: a clientela compra sobretudo os compatíveis da concorrência, os ''clones'', sem que a casa mãe possa reclamar os direitos dos ''made in Taiwan'' (a IBM cometeu a asneira de autorizar a cópia sem licença). Foi a ruptura: a IBM acabou brutalmente com a produção dos PC's e lança o novo standar PS (personal system). Ao mesmo tempo, em vão, instaura processos aos fabricantes de ''clanes'' dos seus PC's para reclamar os seus direitos a título retroactivo.

Um ano e meio mais tarde, pela menos na Europa, o PC da IBM patina, o Mac ascende lentamente e os compatíveis PC portam-se cada vez melhor, melhorando em performances e em preço.

A EVOLUÇÃO DAS UNIDADES CENTRAIS

Existem actualmente quatro referências, cujas características resumimos em baixo.

	Bits de trabalho	Bits de endereçamento	Frequência
8088	8	20	4,77 a 10MHz
8086	16	20	8 a 10MHz
80286	16	32	8 a 12MHz
80386	32	32	12 a 30MHz

Não podemos deixar de referir que a diferença de velocidade entre um 8088 a 4,77MHz e um 80386 a 25MHz é enorme. É, pois, como uma bicicleta em relação a um Lamborghini. Falando de preços: 8086 e 80 88 são iguais; o custo de um 80286 baixou muito e tornou-se acessível, enquanto que 80386 se manteve a preços muito elevados.

A mínima escolha será um 8088 ou 8086 a 8MHz (o de 4,77MHz, muito lento, praticamente desapareceu). A utilização de um PC com um processador 80286 não se justifica para um particular, e até mesmo para muitos profissionais...

Não existe qualquer receio quanto à compatibilidade do software para estes quatro micro-processadores.

A MEMÓRIA RAM

Começou por 128k surgindo, depois, 256k, 512k e finalmente 640k. É o máximo que o MS-DOS sabe gerir. Existem os PC 80386 com 4Mb mas a preço elevado. Com os programas actuais, 512k é o mínimo! Um dos primeiros investimentos ser passar para 640k e isto antes que os circuitos necessários deixem de se encontrar no mercado.

OS MODELOS PC, XT e AT

Trata-se de uma denominação IBM. Com o aparecimento do disco duro havia necessidade de reforçar a designação: os PC são então chamados XT. Com a introdução do 80286, a IBM criou o AT: placa e caixa mais alta, Bios (Rom) modificado e leitor de disquetes de 1,2Mb. O fraco êxito do AT preço e compatibilidade), fez com que surgisse um modelo intermédio, o XT 286: placas e caixas de altura normal, processador 80286, um drive de 1,2Mb e outro, opcional, de 360k. Os compatíveis actuais têm placas de altura normal e são de três tipos:

—os XT com 8088 ou 8086 e drive(s) de 360k ou 720k,
—os 286 com drive 1,2Mb,
—os 386 com drive 1,2mb.
(Nota: um drive 1,2Mb também pode funcionar em 360k).

AS PLACAS PRINCIPAIS

Os primeiros modelos estão bem guarnecidos de ''slots'' (conectores para placas opcionais) pois é preciso uma placa ecrã impressora, uma placa ''controladora de disco'', uma placa multifunções (relógio—calendário + porta série + extensão de memória), etc. Tudo isto torna-se muito caro. Também para fazer baixar os preços, a electrónica miniaturizou,

para integrar o máximo de componentes no circuito impresso principal (a placa mãe), tudo como nas televisões actuais. O mais engraçado é que tudo isso veio trazer fiabilidade à electrónica! Actualmente três slots disponíveis são perfeitamente suficientes.

A EVOLUÇÃO DOS DOS

A versão 2.10 do MS-DOS (apreciada pelos 'craques' das disquetes) foi a primeira, mas com a chegada do disco duro, a versão 3.10 tornou-se uma necessidade. Com a generalização do teclado de 102 teclas surge a versão 3.20.

Com o 286, assistimos à chegada da versão 3.30, e a sintaxe de comando clássicos foi modificada. O volumoso manual do 3..20 reduzido a um pequeno fascículo no qual algumas destas modificações não figuram! Pior, os 80888 trazem o MS-DOS 3.30 e possuem disco duro. Uma reacção bem humana: os numerosos discos duros depressa foram reformatados com o DOS 3.20, excepto para os 286 ou 386 destinados a receber uma extensão de memória (a versão 3.30 era então obrigatória).

Entretanto, assistimos já ao aparecimentos do MS-DOS versão 4 que revolucionou, em parte, o que já existia. Vejamos.

OS TECLADOS

Existiam então os teclados de 85 teclas, com teclas de funções à esquerda, e, à direita, um conjunto de teclas, de deslocação do cursor, ou conjunto numérico. Os Amstrad PC 1512 e PC 1640 conservam-no.

Actualmente a norma é um teclado de 102 teclas, tipo AT, mais largo e mais espaçoso que o anterior, com desdobramento das teclas do cursos e das numéricas, e com as teclas de funções alinhadas no cimo do teclado. Quase todos os sinais de pontuação mudaram de lugar. Porquê? Por causa do sucesso do PC! As operadoras de introdução de dados digitam ''ao quilómetro'' nos terminais de computador. Esses foram progressivamente substituídos por PC's ligados por cabos aos grandes computadores (o que se tornou menos caro). A concepção dos teclados tem pois em consideração uma disposição de teclas idêntica.

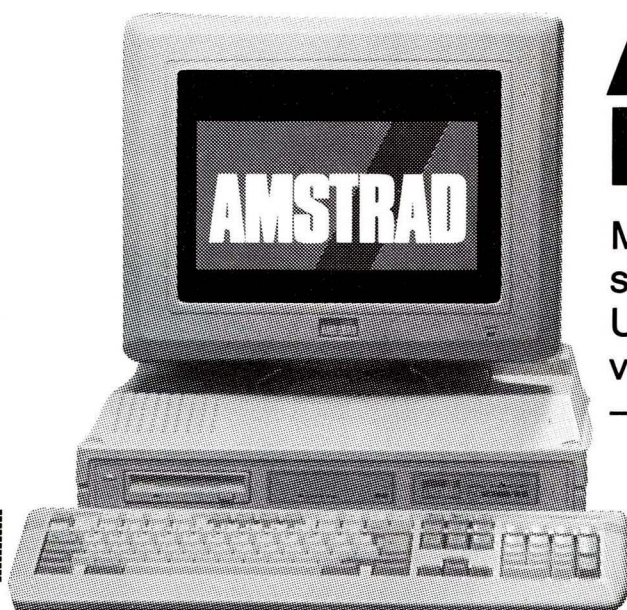
As máquinas de escrever foram progressivamente substituídas por PC's equipados com tratamento de texto. Os construtores conservaram determinadas normas a fim de evitar eventuais inadaptações. O teclado 102 teclas é, pois, uma síntese destas situações.

Os programadores que digitam principalmente os algarismos e as letras maiúsculas abandonam estas 102 teclas Azerty e preferem o bom velho teclado de 85 teclas bloqueado em Caps Lock.

UMA PASSAGEM PELO HORIZONTE DOS PC AMSTRAD

Os PC's Amstrad detêm a melhor relação preço-qualidade para uso pessoal. Em contrapartida, os PC 1512 ou 1640 são pouco integráveis no meio profissional onde se procede com frequência à troca de periféricos: instalar um monitor IBM numa unidade central Epson, um teclado Compaq num Olivetti, etc., o que é impossível com a Amstrad.

DDB NEEDHAM & GUERREIRO

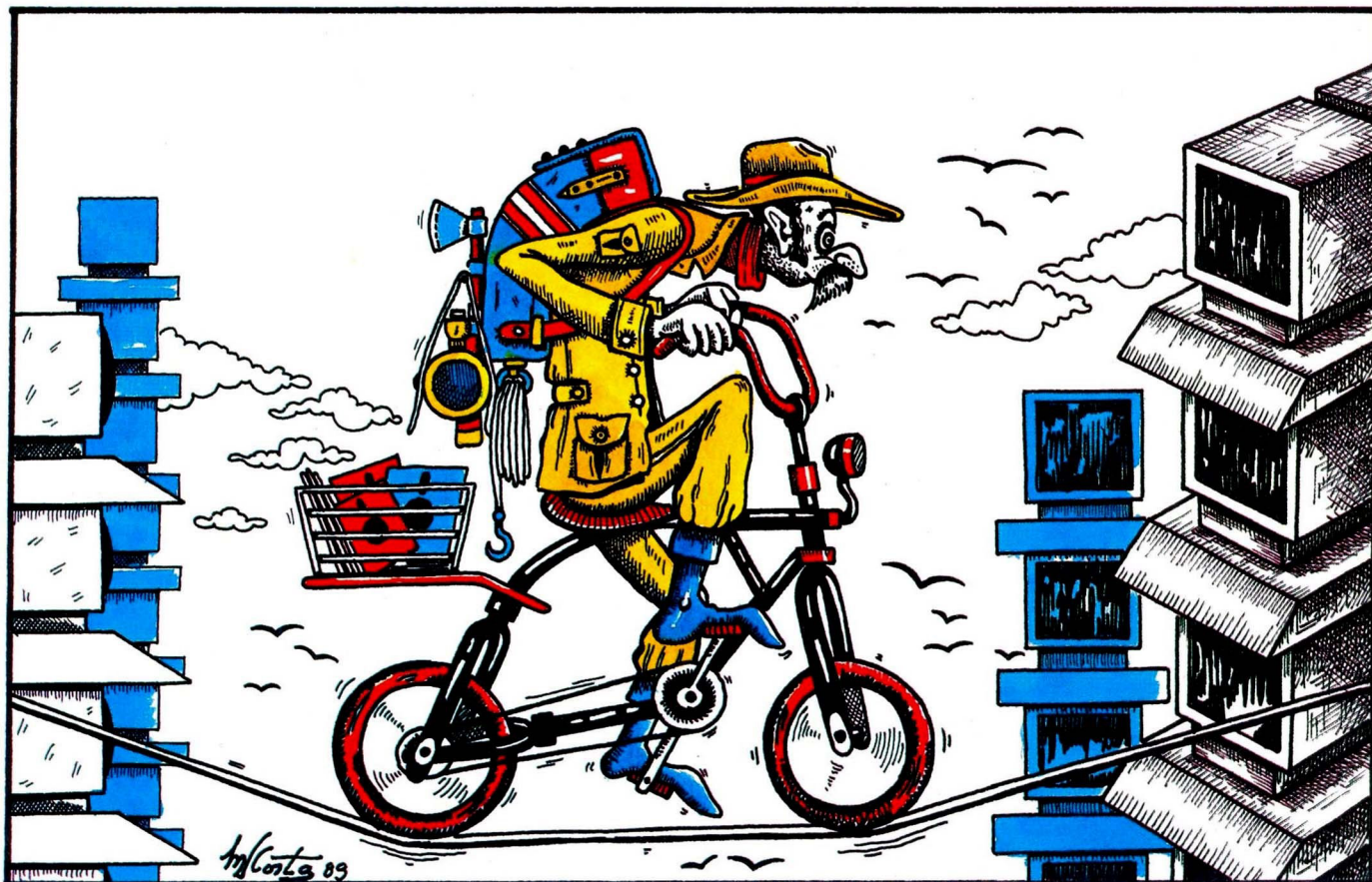
AS CARACTERÍSTICAS FAZEM A DIFERENÇA

AMSTRAD PC 2000

Mais importante que o preço são as características. Uma vez mais a AMSTRAD vai conquistar o mercado – com a série PC 2000.

CARACTERÍSTICAS	AMSTRAD PC 2086	AMSTRAD PC 2286	AMSTRAD PC 2386
Processador Intel	8086 a 8 MHZ	80286 a 12 MHZ	80386 a 20 MHZ
Wait States	—	1/2	0,05
Memória RAM com Verificação de Paridade	640 K	1 Mb	4 Mb
Memória Cache	—	—	64 K a 35 ns
Gestão de Memória	—	LIM 4.0	LIM 4.0
Suporte para Co-Processador Aritmético	8087	80287	80387
Unidade de Diskettes	3 1/2" com 720 K	3 1/2" com 1,4 Mb	3 1/2" com 1,4 Mb
Disco Rígido com Interleave de 1 : 1	30 Mb	40 Mb	65 Mb
Slots de Expansão Livres	3 de 8 bits	5 de 16 bits	5 de 16 bits
Compatível com a Rede Novel 1 Netware	Como Posto de Trabalho	Como «Server»	Como «Server»
Sistema Operativo	MS/DOS 3.3	MS/DOS 4.0	MS/DOS 4.0

CARACTERÍSTICAS COMUNS: Resolução VGA. Saída para unidades externas de diskettes e Streamer. Teclado tipo AT (102 teclas) Português. Rato compatível Microsoft. Segurança do teclado por chave. Windows. GW Basic. Monitores mono e colorido de alta nitidez (DOT PITCH .28).

PCW

EM BUSCA DO FICHEIRO PERDIDO

Enquanto no Locoscript existe uma opção que permite recuperar um documento suprimido, o apagamento de um ficheiro no PCW é catastrófico. O pior é que na abundância de utilitários fornecidos com o CP/M PLUS, não existe nenhum utensílio que permita recuperar um ficheiro que tenha sido sido atirado para o cesto dos papéis por um ERASE intempestivo.

Um pouco de teoria. Contrariamente ao que o seu nome deixa supôr, ERASE (''apaga'' em inglês) não faz desaparecer o ficheiro defunto cujo nome figura sempre na directoria, precedido pelo código E5. É esta marca que o torna invisível aos olhos do utilitário DIR. No caso de o computador necessitar de espaço para alojar dados, estes ficheiros serão esmagados pelos dados recém-chegados. Donde, como conclusão, escrever numa disquete ''com problemas'' põe em perigo a informação escondida que podia aí figurar!

Para recuperar programas ou dados apagados por ERASE, é necessário substituir o código E5 por 00, o número de USER que então lhe será afectado. Na falta de um editor de sector permitindo esta manipulação, propomo-vos um programa rápido e eficaz graças ao qual poderão recuperar os vossos preciosos ficheiros.

Recupera foi publicado recententemente na AMSTRAD CP International, um amigo alemão. Inteiramente traduzido, mostra-se mais eficaz que a versão original a qual só trabalha sobre o leitor A: Esta versão de Recupera reconhece os leitores A: e B: dos PCW 8512 e 9512.

Recupera localiza todos os ficheiros afectados pela infame marca E5 e propõe a sua recuperação. Terminado o trabalho, um simples DIR permitir-vos-à constatar o regresso do ficheiro escondido.

O CPC EM SOCORRO DO PCW

Se encontrar na sua vizinhança um possuidor de CPC 6128 (ou de um 464 com leitor de disquetes) e que possua DISCOLOGY pode dispensar o uso de Recupera em alguns casos. Com efeito, o CPC lê com alguma relutância outros formatos que não o seu; no entanto, DISCOLOGY aceita bastante bem as disquetes do leitor A: do PCW. Graças ao editor é possível aceder aos primeiros sectores da disquete. Assim que os nomes dos ficheiros apareçam, um simples exame do código— legível na extremidade esquerda da janela—vos conduzirá. A transformação dos E5 em ''00'' e a confirmação de actualização serão como um jogo para crianças.

DISQUETES ESTRAGADAS

Ficheiros largamente afectados, com destruição de directoria e outros horrores não mencionáveis poderão ser mais ou menos explorados. Com efeito, enquanto o leitor de disquete recusa aventurar-se em pistas defeituosas, DISCOLOGY sabe ultrapassá-las e encontrar os pedaços de ficheiros-texto espalhados no deserto magnético.

O caminhar errante pelos caminhos de uma disquete é facilitado pela opção ''ASCII unicamente''. DISCOLOGY, saberá mesmo ler, em alguns casos, ficheiros de uma disquete de 720 Kb!

Não esperem sempre ser capazes de recuperar uma folha de cálculo porque sereis confrontados com uma indescritível mescla de dados e fórmulas.

NUNCA DEVE FAZER

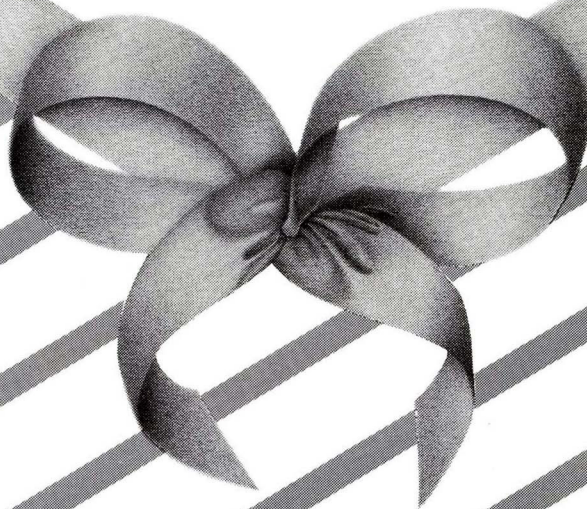
— Formatar uma disquete contendo dados. Estes estarão irremediavelmente perdidos pois toda a organização magnética foi refeita. A directoria que existia foi desfeita; é o vazio sideral.
— Duplicar uma face de disquete B: sobre a outra face da disquete. Contrariamente ao leitor A: que possui uma única cabeça de leitura, o leitor B: possui duas que agarram a disquete simultaneamente como uma sandwich. Virtualmente, a disquete B: é uma única face magnética de grandes dimensões.
— Copiar o conteúdo de uma face sobre a outra gera uma terrível confusão!
— Pelo contrário, nunca deixaremos de insistir na necessidade absoluta de salvaguardar os ficheiros mais importantes! Para que serve ter três cópias do programa MULTIPLAN se a disquete com o seu orçamento anual de oito folhas está totalmente destruída?

Introdução da listagem:

Sob CP/M PLUS, escreva BASIC e carregue em <RETURN>.

```
10 ' RECUPERACAO DE FICHEIROS APAGADOS NO PCW
20 '
30 ' Codigos ESCape:
40 '
50 BIP$=CHR$(7):ESC$=CHR$(27)
60 DOFF$=ESC$+"0":DON$=ESC$+"1"
70 CLS$=ESC$+"E"+ESC$+"H":EFFHT$=ESC$+"d":EFFBAS$=ESC$+"J":PRINT CLS$
80 DEF FNSC$(Z%,S%)=ESC$+"Y"+CHR$(Z%+32)+CHR$(S%+32)
90 CU$=ESC$+"e":CUX$=ESC$+"f"
100 INV$=ESC$+"p":NORM$=ESC$+"q"
110 READSEC=&HF00:WRITESEC=&HF05D:ADRSEC=&HF016
120 DEFINT I
130 '
140 RESTORE 180
150 FOR I%=1 TO 5
160 READ IND%(I%),MG$(I%)
170 NEXT
180 DATA 13,"<S>eguinte       ecuperar    <O>utra disquete    <F>im"
190 DATA 22,"<S>eguinte    <R>ecuperar    <O>utra disquete    <F>im"
200 DATA 30,"<O>utra disquete    <F>im"
210 DATA 18,"Inserir disquete e carregar numa tecla."
220 DATA 26,"Numero do ficheiro a recuperar: "
230 FOR I%=1 TO 5
240 MG$(I%)=MG$(I%)+BIP$
250 NEXT
260 '
270 ' Ecrans
280 '
290 PRINT CUX$;:FOR I=1 TO 13
300 PRINT FNSC$(0,10)CHR$(150);:PRINT STRING$(70,CHR$(154));CHR$(156)
310 PRINT FNSC$(I-1,10)INV$;CHR$(149);SPACE$(70);CHR$(149)
320 NEXT I
330 PRINT FNSC$(12,10)CHR$(147);:PRINT STRING$(70,CHR$(154));CHR$(153)
340 PRINT FNSC$(3,12)"R E C U P E R A C A O   D E   F I C H E I R O S
A P A G A D O S"
350 PRINT FNSC$(5,41)"no P C W"
360 PRINT FNSC$(7,40)STRING$(12,CHR$(154))
370 PRINT FNSC$(10,13);CHR$(164);" Udo RIEGER & C.P.I. 1988. Traduzido e
adaptado por Jose MARCAL"
380 PRINT NORM$;
390 PRINT FNSC$(18,18)"";:INPUT "Em que leitor se encontra a disquete
(A: ou B:) ";LECT$
400 LECT$=LEFT$(LECT$,1):LECT$=UPPER$(LECT$):IF LECT$="A" THEN DRIVE$="00"
410 IF LECT$="B" THEN DRIVE$="01"
420 IF LECT$<>"A" AND LECT$<>"B" THEN PRINT BIP$;CLS$:GOTO 390
430 GOSUB 1320
440 DIM FILE$(128)
450 '
460 ' Procura de dados apagados
470 '
480 DMA=&HF100:J%=0
490 FOR SECT%=0 TO 3
500 CALL READSEC
510 FOR I%=1 TO 16
520 IF HEX$(PEEK(DMA))<>"E5" THEN GOTO 610
530 J%=J%+1
540 FILE$(J%)=STR$(SECT%)+HEX$(DMA)
550 FOR N%=1 TO 11
560 DMA=DMA+1
570 FILE$(J%)=FILE$(J%)+CHR$(PEEK(DMA))
580 NEXT
590 IF INSTR(FILE$(J%),STRING$(11,CHR$(&HE5))) THEN 670
600 DMA=DMA-11
610 DMA=DMA+32
620 NEXT
630 POKE ADRSEC,SECT%+1
640 DMA=&HF100
650 NEXT
660 GOTO 690
670 J%=J%-1
680 PRINT CLS$;
690 IF J%>0 THEN 750
700 PRINT FNSC$(11,20);CLS$;STRING$(53,"_")
710 PRINT FNSC$(12,20);INV$;
```

```
"Nenhum ficheiro apagado figura nesta disquete! ";NORM$;
720 M%=3
730 GOSUB 1230
740 GOTO 930
750 IF J%>1 THEN 780
760 PRINT FNSC$(12,23);"Esta disquete contem 1 ficheiro apagado:";
770 GOTO 790
780 PRINT FNSC$(12,18);J%;"ficheiros apagados foram encontrados nesta
disquete.";
790 Z%=0
800 FOR K%=1 TO J%
810 Z%=Z%+1
820 PRINT FNSC$(Z%+19,35);USING"##";K%;
830 PRINT ".";RIGHT$(FILE$(K%),11)
840 IF Z%=8 THEN 900
850 NEXT
860 PRINT FNSC$(Z%+20,35);STRING$(16," ");
870 M%=2
880 GOSUB 1230
890 GOTO 930
900 IF J%=K% THEN 870
910 M%=1
920 GOSUB 1230
930 A$=INKEY$:A$=UPPER$(A$)
940 IF A$="" THEN 930
950 IF J%=0 THEN 990
960 IF J%=K% THEN 980
970 IF A$="S" THEN PRINT FNSC$(19,0);EFFBAS$;:Z%=0:GOTO 850
980 IF A$="R" THEN 1110
990 IF A$="O" THEN PRINT CLS$:RUN
1000 IF A$="F" THEN PRINT CU$;DON$;CLS$:END
1010 PRINT BIP$:GOTO 930
1020 M%=4
1030 GOSUB 1230
1040 WHILE INKEY$="":WEND
1050 PRINT FNSC$(15,0);EFFBAS$;
1060 ERASE FILE$:GOTO 430
1070 PRINT FNSC$(19,0);EFFBAS$;:Z%=0:GOTO 850
1080 '
1090 '  Recuperacao dos ficheiros
1100 '
1110 M%=5
1120 GOSUB 1230
1130 INPUT "",DON%
1140 PRINT CUX$;
1150 SECT%=VAL(LEFT$(FILE$(DON%),2))
1160 DMA=VAL("&H"+MID$(FILE$(DON%),3,4))
1170 POKE ADRSEC,SECT%
1180 CALL READSEC
1190 POKE DMA,0
1200 CALL WRITESEC
1210 ERASE FILE$
1220 GOTO 430
1230 PRINT FNSC$(30,10);STRING$(65," ")
1240 IF M%=5 THEN 1270
1250 PRINT FNSC$(30,IND%(M%));MG$(M%);
1260 RETURN
1270 PRINT FNSC$(30,IND%(M%));MG$(M%);CU$;
1280 RETURN
1290 '
1300 ' Implementacao dos codigos-maquina
1310 '
1320 MEMORY &HEFFF
1330 RESTORE 1390
1340 FOR ADR=&HF000 TO &HF07F
1350 READ BYTE$:POKE ADR, VAL("&H"+BYTE$)
1360 NEXT
1370 POKE &HF006,VAL("&H"+DRIVE$)
1380 RETURN
1390 DATA C3,34,F0,00,09,00,00,00
1400 DATA 00,00,00,00,0A,00,01,00
1410 DATA 00,00,00,00,0B,00,00,00
1420 DATA 00,00,00,00,0C,00,00,F1
1430 DATA 00,00,00,00,0D,00,00,00
1440 DATA 00,00,00,00,0E,00,00,00
1450 DATA 00,00,00,00,0E,32,11,04
1460 DATA F0,CD,05,00,0E,32,11,0C
1470 DATA F0,CD,05,00,0E,32,11,14
1480 DATA F0,CD,05,00,0E,32,11,1C
1490 DATA F0,CD,05,00,0E,32,11,24
1500 DATA F0,CD,05,00,C9,0E,32,11
1510 DATA 2C,F0,CD,05,00,C9,00,00
1520 DATA 00,00,00,00,00,00,00,00
1530 DATA 00,00,00,00,00,00,00,00
1540 DATA 00,00,00,00,00,00,00,00
```

CARACTERÍSTICAS:
• Processador: 8086
• Memória: 512 K
• Velocidade: 8 MHZ
• Ecrã: CGA
(mono ou cores)
• Um drive 720 K
Você Vai Surpreendê-los!
Natal é o tempo de oferecer e distinguir. A Família e os Amigos. E também os colaboradores da sua Empresa. Talvez eles não estejam à espera. Mas Você vai surpreendê-os. Com o SINCLAIR PC 200, um verdadeiro computador pessoal. Consulte uma loja SOCARTEL e faça com que este Natal seja ainda mais feliz.
SOCARTEL
LOJAS SOCARTEL: LISBOA, OLHÃO, PORTIMÃO, SINES, CARTAXO, LEIRIA, PORTO,
COIMBRA, MONÇÃO, GUIMARÃES, PÓVOA DE VARZIM, CHAVES, BRAGANÇA, PENAFIEL

PCW

«Dr. Logo, suponho...»

NORBERTO TEIXEIRA CONCLUI, NESTA EDIÇÃO, AS SUAS SUGESTÕES DE PROGRAMAS PARA USO DO «DR. LOGO» NO PCW.

PROGRAMA ECRAN

Este programa imprime no ecrã a palavra ''DR. LOGO'', atribuindo valores aleatórios às coordenadas. Necessita ainda do procedimento (locate).

Listagem:

```
to ecran
type char 27 type ''0
ts ct
make ''col 81
make ''lin 30
repeat 80 [locate random :col random :lin pr ''DR.
LOGO]
ern ''col ern ''lin
end

to locate :col :lin
setcursor (list :col :lin)
end
```

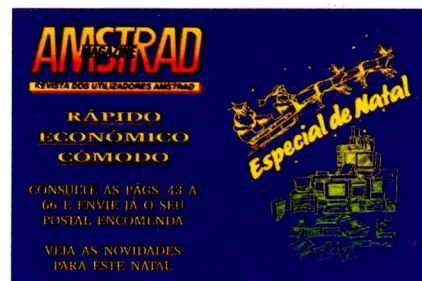
Análise do programa:

O programa começa por apagar a mensagem da última linha do ecrã. Seguidamente limpa todo o ecrã passando o cursor para o canto superior esquerdo (ts ct); em seguida, atribui o valor 81 à variável ''col e 30 à variável ''lin. Na linha a seguir, repete 80 vezes a palavra ''DR. LOGO'' (repare-se como o procedimento 'locate' simplifica este programa!). Termina apagando da memória o nome e o valor das variáveis que tinham sido elegidas. (Pode ser gravado com o nome de ecran, ex: save ''ecran).

Depois de ter introduzido os dois procedimentos (ecran e locate), digite ecran e o programa será executado de imediato.

NOTA: col (abreviatura de coluna), tem o valor 81 para não haver retorno do carro (cr), antes de completar a impressão da palavra ''DR. LOGO'', isto no caso da primitiva random atingir o valor 80.

lin (abreviatura de linha), tem o valor 30, para evitar o scroll.

* se o programa não funcionou, verifique qual a mensagem de erro e digite: edall (o cursor aparecerá no local do erro) para correcção.

PROGRAMA ESQUADRIA

Este programa desenha duas linhas paralelas em redor do ecrã, utilizando para isso o valor decimal do código ascii, correspondente ao caracter gráfico que se deseja imprimir.

Necessita ainda do procedimento (tab).

Listagem:

```
to esquadria
ct type char 27 type ''0
setsplit 2 ss setcursor [0 0]
type char 134 repeat 87 [type char 138] type char
140 pr''
repeat 26 [type char 133 tab 88 pr char 133]
type char 131 repeat 87 [type char 138] pr char 137
end

to tab :col
setcursor (list ;col (last cursor)
end
```

Análise do programa:

O programa começa por apagar o ecrã e a mensagem que aparece na última linha. Seguidamente reserva as duas últimas linhas inferiores para a janela de texto e coloca o cursor no canto superior esquerdo. Na linha seguinte, imprime o caracter gráfico 134, repete 87 vezes horizontalmente o caracter gráfico 138, imprimindo no final da linha o caracter gráfico 140 (pr '') para retorno de linha. Na linha a seguir , repete 26 vezes verticalmente (coluna 0 e coluna 88), o caracter gráfico 133. Na linha que se segue, imprime o caracter gráfico 131, repete 87 vezes horizontalmente o caracter gráfico 138 e conclui com a impressão do caracter gráfico 137.

O programa é executado com a ordem esquadria.

NOTA: Ver no ''MANUAL DO COMPUTADOR'' os códigos ascii dos caracteres gráficos (128 a 159—em valor decimal).

POGRAMA A DESENHAR

Com este programa podemos desenhar no ecrã pressionando as teclas, cujo efeito é apresentado no menu que aparece na parte inferior do ecrã.

Desenhar serve-se do programa anterior ''esquadria'', pois além de melhorar a aparência do ecrã, indica também os limites do espaço que temos para desenhar.

A sua simplicidade passa pela boa distribuição das teclas, estando nelas contidas os principais efeitos. A tecla 8 (rec), serve para que de vez em quando se faça uma reciclagem ao programa, para que este não entre em colapso.

Listagem:

```
to esquadria (introduzir a listagem anterior)
to tab (introduzir a listagem anterior)
to desenhar
menu
inkey
end

to menu
cs esquadria
pr [[Q fd][A bk][O 1t][P rt][1 pu][2 pd][3 pe][4 ht][5
st][6 stop][7 clean][8 rec]]
st
end
to inkey
make ''cmd rc
if cmd = ''q [fd 5]
if cmd = ''a [bk 5]
if cmd = ''o [lt 5]
if cmd = ''p [rt 5]
if cmd = ''1 [pu]
if cmd = ''2 [pd]
if cmd = ''3 [pe bk 5 pd]
if cmd = ''4 [ht]
if cmd = ''5 [st]
if cmd = ''6 [ern ''cmd type [pulse ''c'' para conti-
nuar] stop]
if cmd = ''7 [clean esquadria]
if cmd = ''8 [recycle]
inkey
end
to c inkey
end
```

Análise do programa:

Os procedimentos esquadria e tab, já foram analisados na página anterior; desenhar, manda seguir para o procedimento esquadria, que imprime a esquadria e, imprime agora o menu na parte inferior do ecrã faz aparecer a tartaruga. Regressa novamente a desenhar, que o faz seguir para inkey . Em inkey, rc é atribuído à variável cmd; seguem-se todas as condições e execuções quando se pressionam tais teclas.

O procedimento c (abreviatura de continuar), serve apenas para não termos que escrever desenhar, quando pulsamos a tecla 6.

Ao pulsar a tecla 6, o programa volta ao modo directo, pelo que podemos aqui dar instruções directamente como se o programa não existisse.

Exemplo: fd 50 rt 90

É agora que o procedimento c é útil, pois coloca-nos novamente dentro do programa como se nada tivesse acontecido.

«Dr. Logo, suponho...»

O programa é executado com a ordem desenhar.

Teclas correspondentes as abreviaturas das primitivas usadas no menu e seus efeitos:

Q = >fd — faz a tartaruga avançar.
A = > bk — a tartaruga recua.
O = > lt — a tartaruga gira para a esquerda.
P = > rt — a tartaruga gira para a direita.
1 = > pu — a tartaruga ao deslocar-se não deixa rasto.
2 = > pd — a tartaruga ao deslocar-se deixa rasto.
3 = > pe — a tartaruga ao deslocar-se apaga o rasto.
4 = > ht — esconde a tartaruga.
5 = > st — mostra a tartaruga.
6 = > stop — para a execução do programa.
7 = > clean — apaga os gráficos sem afectar a posição da tartaruga.
8 = > recycle — reorganiza a memória do computador.

NOTA: Para gravar o desenho feito no ecrã fazer: savepic ''nome,ficheiro''.

PROGRAMA TOTOLOTO

Com este programa obtemos chaves com 6 números aleatórios de 1 a 47, que podemos usar para jogar no TOTOLOTO.

Listagem:

```
to totoloto
ct ts
make ''x 0 make ''y 50 make ''z l
pr [* * AMSTRAD * TOTOLOTO * 6 / 47 * *]
repeat 35 [type char 154] pr []
abcdef
end

to abcdef
make (char 47 + :y) :y
make ''y :y + l
if :y = 56 [chave]
abcdef
end

to chave
make ''chave (random 47) + l
make (char 97 + :x) :chave
if (or :b = :a :c = :a :c = :b :d = :a :d = :b
:d = :c :e = :a :e = :b :e = :c :e = :d :f =
:a :f = :b :f = :c :f = :d :f = :e) [chave]
make ''x :x +1
if :x = 6 [imprime]
chave
end

to imprime
if :a = :z [setcursor (list 16 (last cursor)) if
count :a = 1 [type char 32] pr :a]
if :b = :z [setcursor (list 16 (last cursor)) if
count :b = 1 [type char 32] pr :b]
if :c = :z [setcursor (list 16 (last cursor)) if
count :c = 1 [type char 32] pr :c]
if :d = :z [setcursor (list 16 (last cursor)) if
count :d = 1 [type char 32] pr :d]
if :e = :z [setcursor (list 16 (last cursor)) if
count :e = 1 [type char 32] pr :e]
if :f = :z [setcursor (list 16 (last cursor)) if count
:f = 1 [type char 32] pr :f]
make ''z :z +1
if :z = 48 [make ''z 1 continuar?]
imprime
end

to continuar?
repeat 35 [type char 154] pr ''
pr [''c''—CONTINUAR <STOP>—TERMINAR]
recycle
if rc = ''c [totoloto]
end
```

Análise do programa:

Este programa é composto por 5 módulos, ''totoloto'' prepara o ecrã, atribui valor às variáveis x,y e z, imprime o cabeçalho seguido do caracter gráfico 154 e vai para ''abcdef''.

''abcdef'', começa por: 47+y=97 que corresponde à letra 'a', vai repetindo o ciclo até que y seja igual a 56 (daí resulta o nome do procedimento pois 47+51=98 que corresponde ao 'b' e assim sucessivamente até ao 'f'). Para quê este procedimento?—A primitiva random faz sair números aleatórios entre 0 e 1, mas repetindo--os. Assim a cada letra 'abcdef' ser-lhe-á atribuido um valor (primeiro o de y, para que o programa possa prosseguir, depois o de random). O programa segue agora para ''chave'', que muda o valor às variáveis 'abcdef', verifica se alguma delas é igual a outra e segue para ''imprime''.

''imprime'', tem duas funções. A primeira é fazer sair os números por ordem crescente e a segunda acrescentar 1 espaço se a variável só for composta por um número.

Quando 'z' for igual a 48 o programa segue para ''continuar?'' ''continuar'', aqui o programa espera pela tacla ''c'' ou <STOP>, executando ainda uma reciclagem.

O programa arranca escrevendo totoloto.

No procedimento ''chave, a linha—make ''chave (random 47) + 1, pode ser substituída por: make ''chave first shuffle [1 34 45 6 9 27 12 3]. Assim, o programa escolhe somente 6 números dos indicados na lista.

PROGRAMA MORADA

Este programa é uma pequena base de dados para endereços, constando de: Nome, morada, código postal e localidade.

Tem como objectivo final, visualizar tais endereços no ecrã (display) , ou ainda para fazer etiquetas com as direções que o programa contenha (printer). Nesta segunda proposta, os caracteres são impressos em cursivo, pois terão uma melhor apresentação.

Listagem:

```
to morada
(local ''nome ''rua ''cp ''loc)
pedir [Indique o nome ?] ''nome
make ''list1 lput :nome :list1
pedir [Morada (Rua e número) ?] ''rua
make ''list2 lput :rua :list2
pedir [Código postal ?] ''cp
make ''list3 lput :cp :list3
pedir [Localidade ?] ''loc
make ''list4 lput :loc :list4
seguir? [morada] ''t [t para terminar, qualquer tecla
para continuar]
end

to pedir :mensagem :listar
setcursor [0 31] type :mensagem type char 32
make :listar rq
end

to seguir? :processo :final :mensagem
pr :mensagem
if :final = rc [stop]
morada
end

to nova, morada
make ''list1 []
make ''list2 []
make ''list3 []
make ''list4 []
morada
end

to imprimir :list1 :list2 :list3 :list4
if emptyp :list1 [stop]
make ''nome first :list1
pr [Exmo(a) Senhor(a)] pr :nome
make ''rua first :list2
pr :rua
make ''cp first :list3
make ''loc first :list4
pr ''type :cp repeat 5 [type char 32] pr :loc
pr [————————————————————————]
imprimir bf :list1 bf :list2 bf :list3 bf :list4
end

to display
(local ''nome ''rua ''cp ''loc)
ct imprimir :list1 :list2 :list3 :list4
end

to printer
(local ''nome ''rua ''cp ''loc)
ct copyon cursiva
imprimir :list1 :list2 :list3 :list4
n, cursiva copyoff
end
```

```
to cursiva
type char 27 type ''4
end

to n, cursiva
type char 27 type ''5
end
```

Análise do programa:

''morada'', define os nomes para as variáveis locais e serve-se do procedimento ''pedir'', que imprime na última linha do ecrã a mensagem indicativa do pedido de dados, regressa novamente a ''morada'' e cria ''list1. O processo repete-se para ''list2, ''list3,e ''list4, onde serão acrescentados os dados que se vão introduzindo através do procedimento ''seguir?''. Aparece agora em quarto lugar ''nova, morada'', cuja função é criar listas vazias às variáveis: ''list1, ''list2, ''list3 e ''list4. Segue-se depois ''imprimir'', o qual contém a ordem para a saída das 4 list(s). É agora a vez de ''display'' que serve apenas para visualizar as list(s) no ecrã.

''printer'', faz a saída das list(s) para a impressora e para o ecrã, servindo-se de ''cursiva'', que faz activar este modo e ''n,cursiva'' que desactiva o mesmo modo.

Para arrancar com o programa tecle:

''nova morada'' e introduza os dados solicitados. Quando não quiser introduzir mais dados, pode gravá-los. Exemplo:

save ''morada'' (o nome do ficheiro não poderá exceder os 8 caracteres). Sempre que ligar o PCW e introduzir a DR. LOGO, e chamar o programa morada (se foi este o nome que lhe deu), pode acrescentar mais endereços sem apagar os anteriores teclando: morada. Se por outro lado, deseja apagar o conteúdo das listas, bastará teclar de novo: nova, morada.

Exemplo de dois endereços executados por este programa e seu aspecto depois de impressos:

Exmo (a) Senhor (a)
Norberto Manuel Martins Teixeira
Rua D. Sancho I 1.B1 Bargos/Calendario
4760 VILA NOVA DE FAMALICÃO

Exmo (a) Senhor (a)
PUBLINFOR, Publicações e Comércio de Artigos de Informática, S.A.
Avenida da Boavista, 2881-1 andar
4100 PORTO

NOTA: list(s) está em vez de: list1 , list2, list3 e list4

PROGRAMA VER. CONTEÚDO

Este programa imprime no ecrã o conteúdo em valor decimal do endereço da memória, assim como o respectivo código ascii.

Necessita dos procedimentos auxiliares (input e tab)

Listagem:

```
to ver, conteudo
ts ct
input ''end, inicial
input ''end, final
label ''inicio
type :end,inicial tab 7 type ,examine :end,inicial
tab 12 pr char .examine :end,inicial
if :end.inicial = :end.final [ver, conteudo]
make ''end.inical :end.inicial + 1
go ''inicio
end

to input (introduzir a listagem)

to tab (introduzir a listagem)
```

Análise do programa:

O programa começa por limpar todo o ecrã (ts ct). Seguidamente input pede-lhe que indique o valor do endereço onde vai iniciar e terminar a execução do programa. Exemplo:

end, inicial 100
end, final 500

A seguir temos label ''inicio (é para aqui que o programa salta até que seja estabelecida a igualdade das variáveis). Imprime a seguir no ecrã o valor do endereço, o ''peek'' do endereço e o código ascii corresponde ao ''peek''. Na linha a seguir encontra-se a condição de igualdade, segue para a outra linha onde é adicionado o valor 1 à variável end.inicial.. Na linha que se segue o programa salta para label ''início e assim continua até que as variáveis sejam igualadas numericamente.
Observação: O endereço da memória começa em 0 e termina em 65535

PRIMITIVAS DE DR. LOGO

O propósito deste capítulo é o de servir como referência aos utilizadores que não estejam muito familiarizados com esta linguagem.

Gráficos tartaruga

bk n (backward, para trás). Move a tartaruga n passos para trás.

Exemplo: bk 60—move a tartaruga 60 unidades para trás.

fd n (forward, para a frente). Move a tartaruga n passos no sentido da sua orientação.

Exemplo: fd 60—move a tartaruga 60 unidades para a frente.

home. Leva a tartaruga à sua situação inicial: posição [0 0] (centro do ecrã gráfico) e orientação para NORTE (0).

ht (hide turtle, esconder a tartaruga). A tartaruga fica invisível e os desenhos são agora executados a uma maior velocidade.

lt n (left, esquerda). Roda a tartaruga n graus para a esquerda.

Exemplo: lt 60—gira a tartaruga 60 graus para a esquerda.

pd (pen down, descer a caneta). Desce a caneta fazendo com que esta deixe rasto ao mover-se.

pe (pen erase, lápis de apagar). Faz com que a tartaruga apague ao mover-se o que está desenhado.

pu (pen up, subir caneta). Levanta a caneta para que a tartaruga ao mover-se não deixe rasto.

rt (right, direita). Roda a tartaruga n graus para a direita.

Exemplo: rt 60—gira a tartaruga 60 graus para a direita.

seth n (set heading, estabelecer orientação). Gira a tartaruga a partir de Norte.

Exemplo: seth 90—orienta a tartaruga para ESTE.

setpos [n m] (set position, estabelecer posição). Leva a tartaruga ao ponto especificado pelas coordenadas n e m.

Exemplo: setpos [30 50]—leva a tartaruga ao ponto cujas coordenadas x e y sejam 30 e 50 respectivamente.

setx n. Leva a tartaruga ao ponto cuja coordenada x é n, sem modificar a coordenada y.

Exemplo: setx 30—Move a tartaruga no ecrã, até ao ponto em que a coordenada x seja 30.

sety n. Leva a tartaruga ao ponto cuja coordenada y é n, sem modificar a coordenada x.

Exemplo: sety 50—Move a tartaruga no ecrã, até ao ponto em que a coordenada y seja 50.

st (show turtle, mostrar a tartaruga). A tartaruga torna-se visível, se estava invisível.

tf (turtle facts, dados da tartaruga). Escreve no ecrã a informação relativa à tartaruga: posição, orientação e visibilidade.

towards [n m]. Gira a tartaruga de forma a que esta fique apontada ao ponto de encontro das coordenadas n e m.

Exemplo: towards [30 50].

Ecrã gráfico

clean (limpar). Apaga o ecrã gráfico sem afectar a tartaruga.

cs (clear screen, apagar o ecrã). Apaga o ecrã gráfico e restabelece a situação inicial do ecrã: posição [0 0], orientação 0 graus e caneta descida.

dot [n m] (ponto). Desenha um ponto na posição especificada por n e m.

fence (vela). Estabelece um limite que confina a posição da tartaruga e do ecrã gráfico visível.

fs (ful screen, ecrã completo). Assigna o ecrã inteiro para os gráficos.

setscrunch (definir relação de escalas). Estabelece a relação entre as escalas vertical e horizontal do ecrã gráfico (n pode ter qualquer valor compreendido entre 0,1 e 10. O valor inicial é de 0,468.

setsplit n. Especifica o número de linhas de texto que deve haver no ecrã misto.

Exemplo: setsplit 10—reserva as últimas 10 linhas do ecrã para o texto deixando o restante para os gráficos.

sf (screen facts, dados do ecrã). Escreve no ecrã toda a informação relativa ao ecrã gráfico.

ss (split screen, ecrã misto). Estabelece um ecrã para o texto dentro do ecrã grafico.

window (janela). Permite que a tartaruga saia dos limites do ecrã gráfico visível depois de uma ordem wrap ou fence.

wrap (sobrepor borda com borda). Faz com que a tartaruga apareça pelo lado oposto do ecrã quando esta ultrapassa o seu limite.

ecrã de texto

ct (clear text, apagar texto). Apaga o texto da janela onde se encontra o cursor; leva o cursor ao canto superior esquerdo da janela.

cursor, dá como saída uma lista de coordenadas que consiste nos números da coluna e fila da posição actual do cursor na janela de texto.

pr [a b ...]. Escreve no ecrã de texto os objectos especificados; suprime os parêntesis rectos; efectua um retorno do carro depois de escrever o último objecto.

Exemplo: pr [abc]—escreve no ecrã abc e leva o cursor ao princípio da linha seguinte.

setcursor[n m]. Leva o cursor à posição especificada por n e m.

Exemplo: setcursor [35 14]—leva o cursor à coluna 35 e linha linha 14 na janela de texto.14 na janela de texto.

setsplit. Especifica o número de linhas de texto que deve haver no ecrã misto.

Exemplo: setsplit 3—reserva as últimas 3 linhas do ecrã para texto, sendo o restante atribuído à janela gráfica.

show [ab...] (mostrar). Escreve na janela de texto o objecto de entrada; mantem os parêntesis externos das listas; efectua retorno do carro.

ss (split screen, ecrã misto). Estabelece um ecrã para texto dentro do ecrã gráfico.

ts (text screen, ecrã de texto). Assigna o ecrã completo como ecrã de texto.

type [ab...] (escrever). Escreve no ecrã os objectos de entrada; suprime os parêntesis externos das listas; não efectua retorno no carro depois de escrever o último objecto.

Variáveis

erall (erase all, apagar tudo). Apaga todas as variáveis e procedimentos da área de trabalho.
ern [ab...] (erase name, apagar nome). Apaga da área de trabalho as variáveis especificadas.
Exemplo: ern [lado raio]. Apaga as variáveis lado e raio (local ''a ''b). Faz com que a variável ou variáveis de entrada sejam acessíveis somente ao procedimento actual e aos procedimentos invocados por ele.

make ''nome a. Assigna á variável nome o valor a.
Exemplo: make ''lado 50—atribui o valor 50 à variável lado.

nodes. Dá o número de nodes que estão disponíveis no espaço de trabalho.
po ''a (print out, listar). Dá o valor da variável especificada.
Exemplo: Se a variável lado vale 50, po ''lado dará como saída 50.

poall (print out all, listar tudo). Escreve as definições de todos os procedimentos e o valor de todas as variáveis que estão no espaço de trabalho.
pons (print out names, listar nomes). Mostra a lista dos nomes e o valor de todas as varáveis que há no espaço de trabalho.
recycle. Liberta o maior número possível de nodes e reorganiza o espaço de trabalho.
thing ''a (coisa). Dá o valor do objecto citado.
Exemplo: se a variável lado vale 50, thing ''lado dá como saída o número 50.

Operações aritméticas

arctan n. Dá como saída o arco (em graus), cuja tangente é n.
Exemplo: arctan 1—d' 45.

cos n. Dá como saída o cosseno do ângulo n (em graus).
Exemplo: cos 60—dá 0,5.

int n. Dá como saída a parte inteira do número n.
Exemplo: int 1,33—dá 1.

quotient n m. Dá o quociente inteiro da divisão de n por m.
Exemplo: quotient 14 4—dá 3.

random n. Dá um número inteiro aleatório não negativo e menor que o número n.
remainder n m. Dá o resto inteiro da divisão de n por m.
Exemplo: remainder 7 3—dá 1.

round n. Dá como saída o número n arredondando ao inteiro mais próximo.
Exemplo: round 3,33—dá 3, mas round 3,5—dá 4.
sin n. Dá como saída o seno do número n (em graus).
Exemplo: sin 30—dá 0,5.

+ a b ou a+b. Dá a soma de a e b.
Exemplo: + 2 2 dá 4, ou 2+2 dá 4.

— a b ou a-b. Dá a diferença entre b e a.
Exemplo:—10 5 dá 5, ou 10-5 dá 5.

* a b ou a*b. Dá o produto de a e b.
Exemplo: * 4 6 dá 24 ou 4*6 dá 24.

/ a b ou a/b. Dá o resultado de dividir a por b.
Exemplo: / 26 5 dá 5,2 ou 26/5 dá 5,2.

Operações lógicas

and a b. Dá como saída TRUE (verdadeiro) se os valores das expressões lógicas a e b, são ambas verdadeiras; FALSE em qualquer outro caso.
Exemplo: and (3<4) (7>4) dá TRUE.

«Dr. Logo, suponho...»

not a. Dá como saída TRUE se a expressão lógica a é falsa e FALSE em caso contrário.

Exemplo: not (3=4) dá TRUE.

or a b .... Dá como saída FALSE se o valor das expressões lógicas a,b...,são todos falsos e TRUE em qualquer outro caso.

Exemplo: or (3=4) (2=2). dá TRUE, mas or (3=4) (2=1) dá FALSE.

= a b ou a=b. Dá como saída TRUE (verdadeiro) se a é igual a b, e FALSE em caso contrário.

Exemplo: =''casa ''casa dá TRUE, mas 1=2 dá FALSE.

> a b ou a>b. Dá como saída TRUE se a é maior que b, e FALSE em qualquer outro caso.

Exemplo: > 19 20 dá FALSE, mas 20>19 dá TRUE.

> a b ou a>b. Dá como saída TRUE se a é menor que b, e FALSE em qualquer outro caso.

Outras primitivas

contents (conteúdo). Mostra todos os nomes e símbolos reconhecidos por DR.LOGO.

ed ''nome-procedimento. Carrega o procedimento especificado no ecrã do editor, deixando-o listado para a sua edição.

edall (edit all, editar tudo). Carrega todos os procedimentos e variáveis no ecrã do editor e entra em modo de edição.

end. Assinala o final da definição de um procedimento. A palavra end deve figurar sózinha na última linha.

er ''nome-procedimento (erase, apagar). Apaga do espaço de trabalho o procedimento especificado.

erall (erase all, apagar tudo). Apaga todas as variáveis e procedimentos da área de trabalho.

po ''nome-procedimento (print out, listar). Mostra a listagem do procedimento especificado.

poall (print out all, listar tudo). Escreve as definições de todos os procedimentos e o valor de todas as variáveis que se encontram no espaço de trabalho.

pops (print out procedures, listar procedimentos). Dá a lista dos nomes e definições de todos os procedimentos que há no espaço de trabalho.

pots (print out titles, listar títulos). Dá a lista de todos os nomes e entradas dos procedimentos que se encontram no espaço de trabalho.

text ''nome-procedimento. Dá a lista da definição do procedimento.

to. Assinala o início da definição de um procedimento.

Controlo de execução

tye. Termina a sessão de trabalho com DR.LOGO e vai para o sistema operativo (CR/M).

catch ''nome a. Intersecta os erros e situações especiais especificados que ocorram durante a execução da lista de instruções e que tenham sido identificados com o mesmo nome mediante uma instrução throw anterior.

co. Reinicia a execução depois de uma pausa imposta pelo programa.

error. Dá uma lista cujos elementos descrevem o erro mais recente.

go ''palavra. Executa a linha do procedimento actual que está identificada pela etiqueta ''palavra.

if a lista-instruções. Executa a lista de instruções se a expressão lógica a tem o valor verdadeiro(TRUE). Caso contrário, passa à instrução seguinte. Exemplo: if a>b [pr]a é muito grande]]

label ''palavra.Identifica uma lista para que possa ser executada por uma instrução go ''palavra.

op a (op, saída). Converte o objecto a na saída do procedimento e dá por terminada a execução deste.

pause. Suspende a execução do procedimento para permitir a interacção com o intérprete ou com o editor. A execução, reinicia-se com co.

brepeat n [lista-intruções]. Executa a lista de instruções o número de vezes indicado por n.

Exemplo: repeat 4 [fd 30 rt 90]—faz com que a tartaruga avance 30 unidades e gire 90 graus para a direita quatro vezes.

run [lista-instruções]. Executa a lista de instruções especificada.

stop. Detem a execução do programa.

throw ''nome. Funciona em combinação com a instrução catch ''nome para identificar uma situação especial; reenvia o procedimento para a linha que se segue à instrução catch.

Processo de palavras e listas

ascii ''palavra ou ascii [ab...]. Dá o valor ASCII da primeira letra da palavra ou do primeiro caracter da lista.

Exemplo: ascii ''obrigado dá como saída 111.

bf ''palavra ou ascii [a b...] (but first, todos menos o primeiro). Dá como saída todos os caracteres da entrada menos a primeira letra ou menos o primeiro elemento de uma lista.

Exemplo: bf ''trama dá rama; bf [1 2 3] dá [2 3].

bl ''palavra bl [a b...] (but last, todos menos o último). Dá todos os caracteres de palavra menos a última letra ou menos o último elemento da lista.

Exemplo: bl ''tramas dá trama; bl [123] dá [1 2]. char n Dá o caracter cujo código ASCII a n.

Exemplo: char 83 dá a letra s.

count ''palavra ou count [a b...] (count, contar. Dá o número total de letras de ''palavra, ou o número de elementos que há numa lista.

Exemplo: count ''trfs dá o número 4; count [1 2 3] dá 3.

first ''palavra ou first [a b...], (first, primeiro). Dá a primeira letra de uma palavra ou o primeiro elemento de uma lista.

Exemplo: first ''kilo dá k; first [1 2 3] d' 1.

fput ''a''palavra (first put, pôr em primeiro). Dá como saída um objecto já formado, pondo a como primeiro caracter de palavra.

Exemplo: fput ''t ''rama dá trama.

fput a[ab...]. Dá como saída a lista formada, pondo a como primeiro elemento da lista dada.

Exemplo: fput 1 [2 3 4] d' [1 2 3 4].

item n ''palavra ou item n [a b...] (item, elemento). Dá como saída a n letra de uma palavra ou o n elemento de uma lista.

Exemplo: item 4 ''grande dá n; item 2 [1 2 3] dá 2.

last ''palavra ou last [a b...] (last, último). Dá a última letra da palavra ou o último elemento da lista.

Exemplo: last ''dez dá z, last [1 2 3] dá 3.

lc ''palavra (lower case, minúsculas). Dá como saída a palavra de entrada, mas convertendo as maiúsculas a minúsculas.

Exemplo: Norte dá norte.

[list a b...] Dá como saída a lista [a b...].

Exemplo: (list 1 2 3) dá [1 2 3].

lput ''palavra ''palavra 2 (last put, pôr em último). Dá como saída um objecto formado por palavra1 e palavra2, pondo palavra1 no final de palavra2 para obter palavra2 palavra1.

Exemplo: lput ''mente ''nova dá novamente.

lput a [b c...]. Dá como saída a lista formada pondo como primeiro elemento da lista dada.

Exemplo: lput 4[1 2 3] dá [1 2 3 4].

se ''palavra1 ''palavra2 (sentence, frase). Dá como saída uma lista que consta dos elementos palavra1, palavra2,...
Exemplo: se ''um ''dois ''três dá [um dois três].

se [a1 b1...] [a2 b2...].Dá como saída a lista [a1 b1...a2 b2].
Exemplo: se [roxo laranja amarelo] [verde azul] dá a lista [roxo laranja amarelo verde azul].

shuffle [a b...] (baralhar). Dá como saída uma lista que consta dos mesmos elementos que a lista de entrada, mas dispostos por ordem aleatória.
Exemplo: shuffle [1 2 3 4] poderia dar [3 2 4 1].

uc ''palavra (upper case, maiúsculas). Dá como saída a palavra de entrada, mas convertendo as minúsculas maiúsculas.
Exemplo: uc ''Letras dá LETRAS.

word ''palavra1 ''palavra2 (palavra). Dá uma palavra formada com as palavras de entrada.
Exemplo: word ''vice ''versa dá viceversa.

Gestão do disco

chagef ''nome-novo ''nome-antigo (change file, trocar ficheiro). Muda o nome de um ficheiro gravado no disco.
Exemplo: changef ''figura ''estrela dá o nome figura ao ficheiro que antes se chamava estrela.
dir [nome ficheiro] (directory, directório). Dá a lista dos nomes de todos os ficheiros, criados por DR.LOGO no disco.
Exemplo: dir ''a: dá o directório do disco que está na unidade a; dir ''p??????? dá a lista de todos os ficheiros da unidade implícita cujo o nome começa por p.

erasefile ''nome ficheiro. Apaga do directório do disco o ficheiro cujo o nome se especifica.
load ''nome ficheiro. Lê o ficheiro especificado e carrega-o no espaço de trabalho.
save ''nome ficheiro. Grava no disco com o nome especificado o conteúdo do espaço de trabalho.

Captação de dados pelo teclado

rc (read character, lf caracter). Dá como saída o próximo caracter que se introduza pelo teclado.
rl (read list, ler lista). Dá como saída uma que consiste nos próximos caracteres que se introduzam pelo teclado. A entrada termina com um retorno do carro (CR).
rq (road quot, ler literalmente). Dá como saída uma palavra (ou sucessão de palavras), consistindo na linha que se vai introduzindo pelo teclado, termina com (CR).

Controlo da impressora

copyon (cópia activada). Activa o modo 'eco' sendo todos os caracteres que apareçam no ecrã reproduzidos pela impressora.
copyoff (cópia desactivada). Desactiva o modo'eco'.

Exemplo de procedimentos para controlar a escrita da impressora

```
to sublinhar                   to n, sublinhar
type char 27 type ''-1         type char 27 type
end                            end

to negra                       to n, negra
type char 27 type 'È           type char 27 type ''F
end                            end

to letrax2                     to n, letrax2
type char 27 type ''14         type char 27 type ''20
end                            end

to superindices                to n, indices
type char 27 type ''S0         type char 27 type ''T
end                            end
```

Gravação e leitura de imagens (screens)

dirpic [nome ficheiro]. Dá a lista dos ficheiros que contém as screens (imagens do ecrã).
loadpic [nome ficheiro]. Lê o ficheiro de imagem especificado e forma a imagem correspondente no ecrã.
savepic [nome ficheiro]. Grava no disco, com o nome especificando a informação completa sobre a imagem que há no ecrã.

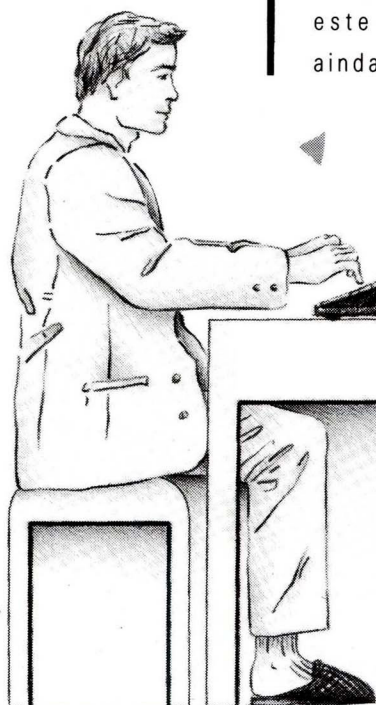
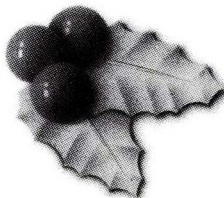
erasepic nome ficheiro. Apaga com o nome especificado o ficheiro de imagem

Exemplo de procedimentos para controlar o ecrã

```
to invideo                     to n, invideo
type char 27 type''p           type char 27 type ''q
end                            end

to invecran                    to n, invecran
type char 27 type ''b0         type char 27 type''bl
type char 27 type''cl          type char 27 type ''c0
end                            end
```

NOTA: invideo, inverte a cor dos caracteres; invecran, inverte a cor do ecrã.

CARACTERÍSTICAS:
• Processador: 8086
• Memória: 512 K
• Velocidade: 8 MHZ
• Ecrã: CGA
(mono ou cores)
• Um drive 720 K
E Para Mim?
No Natal Você costuma oferecer prendas a Familiares e Amigos. Mas, quase nunca se lembra de Si. Este ano, Você pode ter um verdadeiro computador pessoal no seu «sapatinho». Você merece? Você adorava ter? Então, para Si, escolha um SINCLAIR PC 200. Consulte uma loja SOCARTEL e faça com que este Natal seja ainda mais feliz.
SOCARTEL
LOJAS SOCARTEL: LISBOA, OLHÃO, PORTIMÃO, SINES, CARTAXO, LEIRIA, PORTO, COIMBRA, MONÇÃO, GUIMARÃES, PÓVOA DE VARZIM, CHAVES, BRAGANÇA, PENAFIEL

CPC

OS CURSOS DO PROFESSOR ALIGATOR

OS MODOS GRÁFICOS

Para além dos três modos textos 0, 1, e 2 que lhe conhecemos, o CPC dispõe de modos gráficos muito interessantes merecendo um estudo aprofundado.

Examinaremos em detalhe as possibilidades que nos oferecem os caracteres de controlo.

I—INVERSÃO VÍDEO

Permite fazer realçar uma palavra ou frase num texto por permutação de PEN ou PAPER. O texto a imprimir escreve-se com a cor do fundo e o quadro no qual ele se encontra toma a cor do PEN em curso. Obtém-se este efeito fazendo preceder a variável a escrever de um CHR(24); por exemplo:

MODE 0: PEN 3:PRINT CHR=(24) + ''AMSTRAD MAGAZINE'' + CHR=(24)

O retorno à situação normal obtém-se pelo segundo CHR=(24) que anula o efeito de inversão de vídeo.

II—O MODO TRANSPARENTE

Obtém-se com o comando PRINT CHR=(22) + + CHR=(1). A anulação faz-se com PRINT CHR= =(22) + CHR=(0). Este modo autoriza a sobreposição de vários caracteres sem apagar os precedentes.

Á medida que estes últimos são impressos em cores diferentes, podem rivalizar com os famosos sprites da linguagem máquina. Para os fazer mover já é outra história, mas a criação de cenários fixos presta-se bastante bem. Após a utilização do modo transparente, não esqueça de voltar ao modo normal porque este ficará activo, mesmo ao recomeçar a impressão. Em modo transparente a modificação de uma linha utilizando a tecla COPY torna-se um exercício mental de extrema dificuldade.

Como é fastidioso escrever estes comandos diversas vezes, manda a boa técnica defini-los no início do programa. As duas variáveis seguintes figuram na maioria dos programas deste vosso servo:

TR$=chr=(22) + CHR=(1):NR$=CHR$(22) + + CHR$(0)

Outra coisa. Antes de começar qualquer programa novo, é útil introduzir as duas linhas seguintes:

```
1 ON BREAK GOSUB 30 000
30 000 MODE 2:PEN 1:PRINT NR=:END
```

Assim, é possível voltar à listagem sem o incómodo do modo transparente, o PRINT NR= programa-o automaticamente em modo normal. Para fechar este capítulo, é de notar que o modo transparente não pode ser utilizado aquando de uma impressão com TAG. Este comando torna o modo transparente inoperante.

III—O MODO XOR

Aqui, uma incursão à linguagem máquina mostrava-se necessária, mas o espaço disponível a tal não permite. De qualquer forma, saiba que quando um caracter é impresso no modo XOR, os pixels deste adicionam-se aos pixels de fundo. Se o caracter está sobre um fundo uniforme (PAPER 0), ele é impresso tal e qual na cor escolhida. Se se encontra sobre uma parte de um cenário (tipo ilha com palmeiras e sol poente), cada pixel adiciona-se ao que se encontra por debaixo dele, para obter um terceiro pixel de cor diferente.

Até aqui, nada de muito entusiasmante. Mas eis que surge o milagre: no modo XOR, assim que se reimprime

no mesmo local um caracter, ele apaga-se... e restitui o cenário tal e qual. Moral da história: o modo XOR permite deslocar um caracter ao longo do ecrã sem alterar o fundo.

Obtém-se este famoso modo XOR com CHR$(23) + + CHR$(1) e voltar à normalidade por CHR$(23) + + CHRz$(0). Agora, introduza estas poucas linhas e faça RUN:

```
10 MODE 1:PRINT CHR=(23) + CHR(1)
20 FOR H=11 TO 30: FOR G=1 TO 5: LOCATE H,G:PRINT CHR=(143):NEXT G,H
30 PEN 2:FOR H=2 TO 38:LOCATE H,4:PRINT CHR=(251)
40 LOCATE H,4: PRINT CHR=(251): NEXT
```

FATALIDADE, ISTO NÃO ANDA!!!

Não obstante o modo XOR, o nosso personagem impresso duas vezes de seguida no mesmo sítio fica visível sobre toda a linha e, para além disso, apagou o cenário à sua passagem. Com efeito, é necessário saber que em BASIC, o modo XOR só funciona associado ao modo gráfico, enquanto que, neste caso, estamos em modo texto. Para solucionar este problema, é necessário utilizar o comando TAG e TAGOFF. Substitua as linhas 30 e 40 pelo que se segue e faça de novo RUN.

```
30 PLOT 700,700,2:TAG:FOR H=32 TO 618 STEP 16
40 MOVE H,336: PRINT CHR=(251);:FOR T=1 TO 100:NEXT
50 MOVE H,336: PRINT CHR=(251);:NEXT:TAGOFF
```

Desta vez alcança-se o resultado esperado. O nosso pequeno homem desloca-se frente ao muro sem o apagar. Repare que durante a sua passagem, a cor transforma-se devido ao efeito da adição dos pixels evocada mais acima.

Esta pequena listagem merece alguns comentários:

Linha 10: Passagem ao modo XOR.

Linha 20: Criação do cenário (muito estilizado).

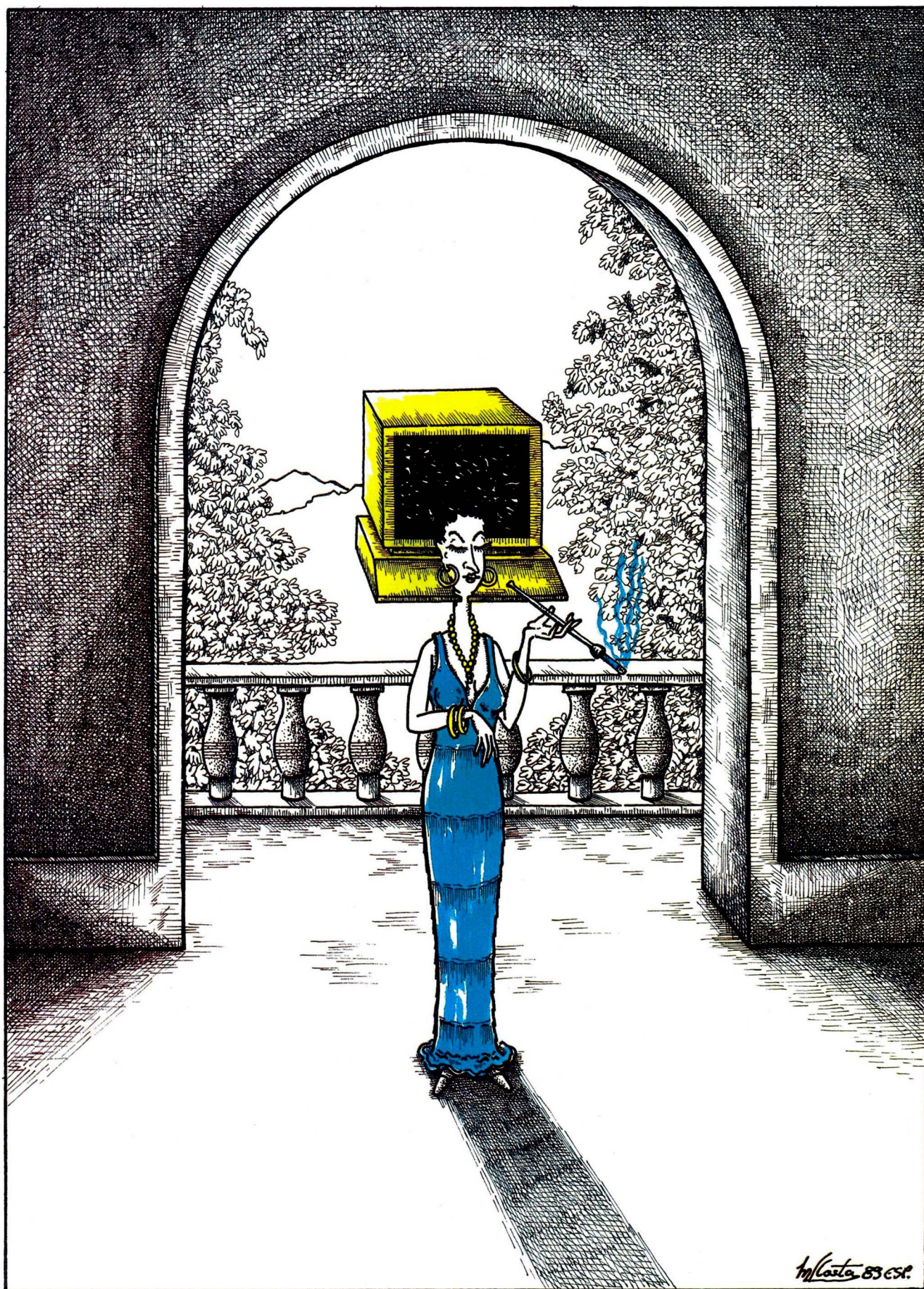
Linha 30: O PLOT define a cor. TAG assegura a passagem a modo gráfico. O ciclo define um deslocamento equivalente a uma casa em modo 1.

Linha 40: Primeira impressão do personagem mais pausa

Linha 50: Segunda impressão que tem por efeito apagar o personagem, depois salto para uma nova posição.

De notar que esta listagem, embora realizada para um 464, corre muito bem num 6128. No entanto, neste último, o comando MOVE admite quatro parâmetros, sendo o terceiro a cor e o quarto o modo gráfico. Assim, poderíamos, para o 6128, ultrapassar o PLOT e a passagem a XOR escrevendo MOVE X,Y,2,1.

E, por hoje, fiquemos por aqui.

M.Costa 83 ESP.

1.º PRÉMIO

FRANCISCO MIGUEL DE O. SPINOLA
PÓVOA DE SANTA IRIA

Estamos perante um programa com qualidade e bem elaborado, é um processador de texto bastante bom e com todas as capacidades dos processadores de texto comuns. Permite escrever textos com um máximo de 750 linhas, possui instruções, opcção de tabela ASCII, tratamento dos ficheiros, como criação, abertura e anulação, impressão e para além disso um pequeno conjunto de teclas de função que facilitam e apressam o processamento.

É na realidade um bom programa e sem erros, pelo menos detectáveis no nosso teste.

```
{ CABECALHO }

program EDITOR(input,output,arq_texto);

{ BLOCO DE DECLARACOES }

const RETURN    =#13;
      maxlinha   =750; { MAXIMA LINHA }
      maxcol    =78;   { MAXIMA COLUNA }
      up        =true;
      down      =false;
type tipo_de_lista=array[1..maxlinha,1..maxcol]of char;
var lista                :tipo_de_lista;
    c,d,x,y,linha,col,
    max_col_escrita,
    min_col_escrita      :integer;
    drive                :string[2];
    st                   :string[78];
    fich1,fich2          :string[12];
    functionkey,
    gravacao,
    pass,
    tipo_de_escrita      :boolean;
    opcao,carac          :char;
    arq_texto            :text;
{FUNCOES}

{DEVOLVE O VALOR DA MAXIMA LINHA ESCRITA E RESPECTIVA COLUNA}
function  max_linha_escrita:integer;
begin
  for c:=750 downto 1 do
   begin
     for d:=78 downto 1 do
      begin
        if lista[c,d]<>' ' then begin
                                  max_linha_escrita:=c;
                                  max_col_escrita:=d;
                                  exit;
                                end;
      end;
   end;
   max_col_escrita:=1;
   max_linha_escrita:=1;
end;
{DEVOLVE O VALOR DA PRIMEIRA LINHA ESCRITA E RESPECTIVA COLUNA}
function min_linha_escrita:integer;
begin
  for c:=1 to 750 do
   begin
     for d:=1 to 78 do
      begin
        if lista[c,d]<>' ' then begin
                                  min_linha_escrita:=c;
                                  min_col_escrita:=d;
                                  exit;
                                end;
      end;
   end;
   min_col_escrita:=1;
   min_linha_escrita:=1;
end;

{PROCEDIMENTOS}

{REALIZA A APRESENTACAO DO PROGRAMA}
procedure apresentacao;
begin
  clrscr;
  gotoxy(2,5);
  write('[illegible]');
  gotoxy(2,6);
  write('[illegible]');
  gotoxy(2,7);
  write('[illegible]');
  gotoxy(2,8);
  write('[illegible]');
  gotoxy(2,9);
  write('[illegible]');
  textcolor(brown);
  gotoxy(30,17);write('( ) Copyright 1988');
  gotoxy(18,18);write('por ');
  textcolor(white);write('Francisco Miguel de Oliveira Spinola');
  textcolor(white);
  gotoxy(27,17);write('V1');
  gotoxy(31,17);write('C');
  textcolor(brown);
  gotoxy(1,24);write('Prima ');
  textcolor(white);
  write(chr(17),'─┘ ');
  textcolor(brown);
  write(' para continuar ... ');
  textcolor(white);
  repeat until keypressed;
end;
 {JANELAS}
 procedure janela(a,b,w,d,e,f:char;cimaesq,cimadir,latcima,latbaixo:integer);
 begin
   for c:=(cimaesq+1) to (cimadir-1) do
    begin
      gotoxy(c,latcima);write(e);
      gotoxy(c,latbaixo);write(e);
    end;
  for c:=(latcima+1) to (latbaixo-1) do
   begin
     gotoxy(cimaesq,c);write(f);
     gotoxy(cimadir,c);write(f);
   end;
  gotoxy(cimaesq,latcima);write(a);
  gotoxy(cimadir,latcima);write(b);
  gotoxy(cimaesq,latbaixo);write(w);
  gotoxy(cimadir,latbaixo);write(d);
 end;

 {MENSAGEM PARA ESPERAR}
 procedure esperar;
 begin
   clrscr;
   textbackground(white);
   textcolor(white);
   gotoxy(27,12);write('   Espere um momento sff     ');
   textcolor(black);
   janela('┌','┐','└','┘','─','│',25,56,11,13);
   gotoxy(55,12);
   textcolor(white);
 end;

{MENSAGEM PARA INTRODUZIR A DISKETTE}
procedure quadro_1;
begin
  clrscr;
  textbackground(white);
  textcolor(white);
  gotoxy(32,12);write('Introduza a diskette');
  gotoxy(32,13);write('e prima uma tecla');
  textcolor(black);
  janela('┌','┐','└','┘','─','│',28,55,10,15);
  janela('┌','┐','└','┘','─','│',29,54,11,14);
  gotoxy(50,13);
  repeat until keypressed;
  textcolor(white);
end;

{INTRODUCAO DO NOME DO FICHEIRO}
procedure quadro_2;
begin
    clrscr;
    textbackground(white);
    textcolor(white);
    gotoxy(30,12);write('Nome do ficheiro:');
    textcolor(black);
    janela('┌','┐','└','┘','─','│',27,61,10,14);
    janela('┌','┐','└','┘','─','│',28,60,11,13);
    textcolor(white);
    gotoxy(47,12);
    repeat
     buflen:=12;
     delete(fich1,1,12);
     read(fich1);
    until fich1>'
end;

 {MENSAGEM DE ERRO}
 procedure quadro_3;
 begin
    clrscr;
    write(^g);
    textbackground(white);
    textcolor(white);
    gotoxy(34,12);write('Erro de I/O');
    textcolor(black);
    janela('┌','┐','└','┘','─','│',30,48,10,14);
    janela('┌','┐','└','┘','─','│',31,47,11,13);
    gotoxy(53,12);
    textcolor(white);
    gotoxy(29,16);write('F1-            ESC-       ');
    textcolor(brown);
    gotoxy(32,16);write('Continuar');
    gotoxy(46,16);write('Parar');
    textcolor(white);
    repeat
     d:=1;
     repeat
       read(kbd,opcao);
     until opcao=#27;
     if keypressed then begin
                          read(kbd,opcao);
                          if opcao<>';' then d:=0;
                        end;
    until d<>0;
 end;

 {PROCEDURE AUXILIAR}
 procedure caixa;
 begin
   clrscr;
   textbackground(white);
   textcolor(white);
   gotoxy(21,12);write('O ficheiro:');
   write(fich1:13);
   gotoxy(46,12);write('foi');
   textcolor(black);
   janela('┌','┐','└','┘','─','│',19,61,11,13);
   textcolor(white);
 end;

 {MENSAGEM SE QUER GRAVAR}
 procedure quadro 4;
 begin
   clrscr;
   textbackground(white);
   textcolor(white);
   gotoxy(16,12);write(fich1:13,' nao foi gravado.Gravar(S/N)?');
   textcolor(black);
   janela('┌','┐','└','┘','─','│'.14,60.11,13);
   textcolor(white);
   repeat
    gotoxy(57.12);
    read(kbd,opcao);
  until opcao in['s'.'S','N'.'n'];
  textcolor(white);
 end;

 {MENU INICIAL DE OPCOES}
 procedure quadro;
 begin
 clrscr;
 textbackground(white);
 textcolor(brown);
 gotoxy(28,8); write('[illegible]');
 gotoxy(28,9); write('[illegible]');
 gotoxy(28,10);write('[illegible]');
 gotoxy(28,11);write('[illegible]');
 gotoxy(28,12);write('[illegible]');
 gotoxy(28,13);write('[illegible]');
 gotoxy(28,14);write('[illegible]');

 gotoxy(28,16);write('[illegible]');
 gotoxy(28,17);write('[illegible]'),
 gotoxy(28,18);write('[illegible]');
 textcolor(black);
 gotoxy(28,16);write('┌───┐');
 gotoxy(28,17);write('│');
 gotoxy(32,17);write('│');
 gotoxy(28,18);write('└───┘');
 gotoxy(28,8); write('┌─────  Opçoes ');
 gotoxy(43,8);write(' ─────┐');
 for c:=9 to 13 do
   begin
    gotoxy(28,c);write('│');
    gotoxy(49,c);write('│');
   end;
```

```
  gotoxy(28,14);write('└──────────────────┘');
  textcolor(brown);
  gotoxy(32,10);write(' riar ficheiro');
  gotoxy(32,11);write(' er ficheiro');
  gotoxy(32,12);write(' air');
  textcolor(white);
  gotoxy(32,10);write('C');
  gotoxy(32,11);write('L');
  gotoxy(32,12);write('S');
  repeat
    opcao:=' ';
    gotoxy(30,17);
    read(kbd,opcao);
  until opcao in ['l','L','C','c','s','S'];
end;

{QUADRO PRINCIPAL DO EDITOR}
procedure quadro_principal;
begin
  clrscr;
  textbackground(white);
  textcolor(brown);
  janela('█','█','█','█','█','█',1,80,1,23);
  textcolor(black);
  for c:=1 to 80 do
  begin
    gotoxy(c,1);write(' ');
    gotoxy(c,23);write(' ');
  end;
  for c:=2 to 22 do
  begin
    gotoxy(1,c);write(' ');
    gotoxy(80,c);write(' ');
  end;
  gotoxy(37,1);write('Linha:');
  gotoxy(48,1);write('Coluna:');
  gotoxy(61,1);write('Escrita:');
  gotoxy(1,1);write('┌');
  gotoxy(80,1);write('┐');
  gotoxy(1,23);write('└');
  gotoxy(80,23);write('┘');
  for c:=2 to 22 do
    begin
      gotoxy(1,c);write('│');
      gotoxy(80,c);write('│');
    end;
  for c:=2 to 79 do
  begin
    gotoxy(c,23);write('─');
  end;
  highvideo;
  gotoxy(5,24);
  write('F1        F2                F3      F4       F5        F6           ESC     ');
  normvideo;
  textbackground(black);
  textcolor(brown);
  gotoxy(7,24);write('-Ajuda ');
  gotoxy(16,24);write('-Tabela ascii ');
  gotoxy(32,24);write('-Ler ');
  gotoxy(39,24);write('-Criar ');
  gotoxy(48,24);write('-Gravar ');
  gotoxy(58,24);write('-Imprimir ');
  gotoxy(71,24);write('-Sair ');
end;

{QUADRO DA TABELA ASCII}
procedure quadro_ascii;
begin
  textbackground(white);
  textcolor(brown);
  janela('█','█','█','█','█','█',27,49,3,20);
  textcolor(black);
  for c:=4 to 19 do begin
                      gotoxy(27,c);write('│');
                      gotoxy(49,c);write('│');
                    end;
  gotoxy(27,3); write('┌─────────────────────┐');
  gotoxy(27,20);write('└─────────────────────┘');
  textcolor(black);
  gotoxy(30,3);write('Dec Car   Dec Car');
  for c:=4 to 19 do begin
                      textcolor(black);
                      gotoxy(38,c);write('│');
                    end;
  textcolor(white);
  gotoxy(24,22);write('PgUp      PgDn       ESC     ');
  textcolor(brown);
  gotoxy(28,22);write('-Cima');
  gotoxy(38,22);write('-Baixo');
  gotoxy(48,22);write('-Sair');
end;

{ ESCRITA DOS CODIGOS ASCII}
procedure escrever_ascii(maximo,minimo:integer);
begin
  textcolor(white);
  d:=4;
  window(28,4,37,19);
  clrscr;
  window(39,4,48,19);
  clrscr;
  window(1,1,80,25);
  for c:=minimo to (maximo-16) do
   begin
     gotoxy(30,d);write(c);
     if c<>7 then begin
                    gotoxy(34,d);
                    write(chr(c));
                  end;
     d:=d+1;
   end;
  d:=4;
  for c:=(minimo+16) to maximo do
   begin
     gotoxy(40,d);write(c);
     gotoxy(44,d);write(chr(c));
     d:=d+1;
   end;
end;

{AVANCAR OU RECUAR NA TABELA ASCII}
procedure tabela_ascii;
var maximo,minimo:integer;
begin
  clrscr;
  quadro_ascii;
  maximo:=31;
  minimo:=0;
  escrever_ascii(maximo,minimo);
  repeat
  repeat
    read(kbd,opcao);
  until opcao=#27;
  if keypressed then
   begin
     read(kbd,opcao);
     case opcao of
              'I':begin
                    if maximo<224 then
                     begin
                       maximo:=maximo+32;
                       minimo:=minimo+32;
                     end
                    else
                    begin
                      maximo:=31;
                      minimo:=0;
                    end;
                    escrever_ascii(maximo,minimo);
                  end;
              'Q':begin
                    if minimo=0 then
                     begin
                       minimo:=224;
                       maximo:=255;
                     end
                    else
                      begin
                        minimo:=minimo-32;
                        maximo:=maximo-32;
                      end;
                      escrever_ascii(maximo,minimo);
                    end;
             end;{ Case }
   end;{ If }
  until opcao=#27;
  clrscr;
end;

{ANULAMENTO DA TABELA}
procedure anular;
begin
  for c:=1 to maxlinha do
   begin
     for d:=1 to maxcol do
       begin
         lista[c,d]:=' ';
       end;
   end;
   linha:=2;
   col:=2;
   y:=1;
   x:=1;
end;

{ESCRITA DA POSICAO AONDE SE ENCONTRA O CURSOR}
procedure posicoes;
begin
  textbackground(white);
  textcolor(black);
  gotoxy(43,1);write(x:3);
  gotoxy(55,1);write(y:2);
  textcolor(white);
end;

{ESCRITA DO NOME DO ARQUIVO E DRIVE ACTUAL}
procedure nome_do_arquivo;
begin
  textbackground(white);
  textcolor(black);
  gotoxy(6,1);write(' ',drive:2);
  gotoxy(9,1);write(fich1);
  textcolor(white);
end;

{ AVANCA PARA ALINHA SEGUINTE OU RECUA PARA A ANTERIOR}
procedure scroll_dados(scroll:boolean);
begin
  window(2,2,79,22);
  if scroll then
   begin
     gotoxy(1,1);
     delline;
     gotoxy(1,20);
     write(lista[x]);
   end
  else
   begin
     gotoxy(1,1);
     insline;
     gotoxy(1,1);
     write(lista[x]);
   end;
  window(1,1,80,25);
end;

{AVANCA O CURSOR UMA POSICAO PARA CIMA}
procedure cursor_cima;
begin
   if x>1 then
    begin
      x:=x-1;
      if linha=2 then scroll_dados(down);
      if linha>2 then linha:=linha-1;
    end;
end;

{AVANCA O CURSOR UMA POSICAO PARA BAIXO}
procedure cursor_baixo;
begin
  if x<maxlinha then
   begin
     if linha=22 then begin
                        scroll_dados(up);
                        gotoxy(2,22);write(lista[x+1]);
                      end;
     if linha<22 then linha:=linha+1;
     x:=x+1;
     if carac=RETURN then
      begin
        y:=1;
        col:=2;
      end;
   end;
end;

{AVANCA O CURSOR UMA POSICAO PARA A DIREITA}
procedure cursor_direita;
begin
  if y<maxcol then
   begin
     col:=col+1;
     y:=y+1;
   end;
end;

{AVANCA O CURSOR UMA POSICAO PARA A ESQUERDA}
procedure cursor_esquerda;
begin
  if y>1 then
   begin
     col:=col-1;
     y:=y-1;
   end;
end;

{AVANCA PARA A PAGINA SEGUINTE OU RECUA PARA A PAGINA ANTERIOR}
procedure scroll_Pagina(scroll:boolean);
begin
  window(2,2,79,22);
  clrscr;
  window(1,1,80,25);
  if scroll=up then
   begin
     if x<22 then begin
                    d:=21;
                    x:=1;
                    linha:=2;
                    for c:=22 downto 2 do
                     begin
                       gotoxy(2,c);write(lista[d]);
                       d:=d-1;
                     end;
                  end;
     if x>21 then begin
                    d:=x-(linha-2);
                    x:=x-20;
                    if d<21 then begin
```

```
                        d:=21;
                        x:=linha-1;
                      end;
                for c:=22 downto 2 do
                 begin
                   gotoxy(2,c);write(lista[d]);
                   d:=d-1;
                 end;
              end;
end;
if scroll-down then
  begin
    if x>730 then begin
                    d:=730;
                    x:=750;
                    linha:=22;
                    for c:=2 to 22 do
                     begin
                       gotoxy(2,c);write(lista[d]);
                       d:=d+1;
                     end;
                  end;
    if x<731 then begin
                    d:=x+(22-linha);
                    x:=x+20;
                    if d>730 then begin
                                    d:=730;
                                    x:=750-(22-linha);
                                  end;
                    for c:=2 to 22 do
                     begin
                       gotoxy(2,c);write(lista[d]);
                       d:=d+1;
                     end;
                  end;
  end;
end;

procedure listar_texto;
begin
  gotoxy(2,linha);write(lista[x]);
  d:=x+(22-linha);
  gotoxy(2,22);write(lista[d]);
end;

{RETIRA UMA LINHA DA TABELA}
procedure retirar_linha;
begin
  window(2,2,79,22);
  gotoxy(1,linha-1);
  delline;
  for c:=x to 749 do lista[c]:=lista[c+1];
  for c:=1 to 78 do lista[750,c]:=' ';
  window(1,1,80,25);
  listar_texto;
end;

{INSERE UMA LINHA NA TABELA}
procedure inserir_linha;
begin
  window(2,2,79,22);
  gotoxy(1,linha-1);
  insline;
  for c:=750 downto (x+1) do lista[c]:=lista[c-1];
  for c:=1 to 78 do lista[x,c]:=' ';
  window(1,1,80,25);
end;

{FAZ A INSERCAO DE ESCRITA}
procedure escrita_inserida;
begin
  for c:=78 downto y+1 do lista[x,c]:=lista[x,c-1];
  lista[x,y]:=carac;
  d:=y;
  for c:=col to (maxcol+1) do
   begin
     gotoxy(c,linha);write(lista[x,d]);
     d:=d+1;
   end;
end;

{LISTA O TIPO DE ESCRITA ACTUAL inserida ou sobreposta}
procedure listar_escrita;
begin
  textbackground(white);
  textcolor(black);
  gotoxy(69,1);
  if tipo_de_escrita=True then write('Inserir ')
   else
    write('Sobrepor');
  textcolor(white);
end;

{APAGA O CARACTER ATRAS DO CURSOR}
procedure apagar_tras;
begin
  if (y=1)and(x>1)then begin
                         x:=x-1;
                         linha:=linha-1;
                         for c:=78 downto 1 do
                          begin
                            if lista[x,c]<>' ' then
                             begin
                               col:=c+1;
                               y:=c;
                               exit;
                             end;
                          end;
                         y:=1;
                         col:=2;
                         exit;
                       end;
  if (y>1)then
   begin
     for c:=y to 78 do lista[x,c-1]:=lista[x,c];
     lista[x,78]:=' ';
     y:=y-1;
     col:=col-1;
     d:=y;
     for c:=col to (maxcol+1) do
      begin
        gotoxy(c,linha);write(lista[x,d]);
        d:=d+1;
      end;
   end;
end;

{APAGA O CARACTER A FRENTE DO CURSOR}
procedure apagar_frente;
begin
  for c:=y to 77 do lista[x,c]:=lista[x,c+1];
  lista[x,78]:=' ';
  d:=y;
  for c:=col to (maxcol+1) do
   begin
     gotoxy(c,linha);write(lista[x,d]);
     d:=d+1;
   end;
end;

{AVANCA O CURSOR DE 6 EM 6 POSICOES}
procedure tabulador;
begin
  if y>72 then begin
                 y:=78;
                 col:=79;
               end
   else
     begin
       y:=y+6;
       col:=col+6;
     end;
end;

{COLOCA O CURSOR NA PRIMEIRA COLUNA (Home)}
procedure cursor_principio;
begin
  col:=2;
  y:=1;
  gotoxy(linha,col);
end;

{COLOCA O CURSOR NA ULTIMA COLUNA ESCRITA (End)}
procedure cursor_Fim;
begin
  for c:=maxcol downto 1 do
   begin
     if lista[x,c]<>' ' then begin
                               if (c>1)and(c<maxcol)then
                               begin
                                 y:=c+1;
                                 col:=c+2;
                               end
                                else
                                  begin
                                    y:=c;
                                    col:=c+1;
                                  end;
                               gotoxy(col,linha);
                               exit;
                             end;
   end;
end;

{SUBSTITUI CARACTERES DA TABELA POR NOVOS CARACTERES}
procedure substituir;
var antigo,novo:char;
begin
  repeat
    clrscr;
    textbackground(white);
    gotoxy(27,12);write('Qual o caracter a mudar:');
    textcolor(black);
    janela('┌','┐','└','┘','─','│',25,53,11,13);
    gotoxy(51,12);buflen:=1;
    textcolor(white);
    read(antigo);
    clrscr;
    gotoxy(29,12);write('Qual o novo caracter:');
    textcolor(black);
    janela('┌','┐','└','┘','─','│',27,52,11,13);
    gotoxy(50,12);buflen:=1;
    textcolor(white);
    read(novo);
    clrscr;
    esperar;
    for c:=1 to max_linha_escrita do
    for d:=1 to 78 do if lista[c,d]=antigo then lista[c,d]:=novo;
    clrscr;
    textcolor(black);
    gotoxy(36,11);write('┌────────────────┐');
    gotoxy(36,12);write('│');
    gotoxy(53,12);write('│');
    gotoxy(36,13);write('└────────────────┘');
    textcolor(white);
    gotoxy(38,12);write('Repetir (S/N)?');
    repeat
      read(kbd,opcao);
    until opcao in ['s','S','n','N'];
  until (opcao='n')or(opcao='N');
  clrscr;
end;

{MUDA O CURSOR PARA UMA LINHA ENTRE 1 E 750}
procedure mudar_de_linha(var numero:integer);
begin
  clrscr;
  textbackground(white);
  textcolor(white);
  gotoxy(22,12);write('Qual o numero da linha(max.750)?');
  textcolor(black);
  janela('┌','┐','└','┘','─','│',20,58,11,13);
  textcolor(white);
  repeat
    {$I-}
    gotoxy(54,12);write('   ');
    gotoxy(54,12);
    buflen:=3;
    numero:=0;
    read(numero);
    {$I+}
  until (ioresult=0)and(numero>0)and(numero<751);
  clrscr;
  if numero<730 then begin
                       linha:=2;
                       col:=2;
                       y:=1;
                     end
               else
                begin
                 linha:=(numero-730)+2;
                 col:=2;
                 y:=1;
                end;
end;

{APAGA OS CARACTERES ATRAS DO CURSOR}
procedure apagar_atras_cursor;
begin
  for c:=1 to y-1 do lista[x,c]:=' ';
  for c:=2 to (col-1) do
   begin
     gotoxy(c,linha);write(lista[x,c-1]);
   end;
end;

{APAGA OS CARACTERES A FRENTE DO CURSOR}
procedure apagar_frente_cursor;
begin
  for c:=(y+1) to maxcol do lista[x,c]:=' ';
  for c:=(col+1) to (maxcol+1) do
   begin
     gotoxy(c,linha);write(lista[x,c-1]);
   end;
end;

{MOVE O CURSOR PARA O INICIO DA PALAVRA QUE SE LHE SEGUE}
procedure palavra_frente;
begin
  for c:=(y+1)to 78 do
   if (lista[x,c-1]=' ')and(lista[x,c]<>' ')then
    begin
      col:=c+1;
      y:=c;
      exit;
    end;
end;

{MOVE O CURSOR PARA O INICIO DA PALAVRA QUE O ANTECEDE}
procedure palavra_atras;
begin
if y>1 then
begin
 if y>3 then
  begin
   for c:=(y-1) downto 2 do
   begin
    if(lista[x,c]<>' ')and(lista[x,c-1]=' ') then
```

```
      begin
        col:=c+1;
        y:=c;
        exit;
      end;
    if(c=2)and(lista[x,1]<>' ')then begin
                                      col:=2;
                                      y:=1;
                                      exit;
                                    end;
   end;
  end;
  if y<4 then begin
               if lista[x,1]<>' ' then begin
                                          col:=2;
                                          y:=1;

                                        end;
               if (lista[x,1]=' ')and(lista[x,2]<>' ') then
                begin
                  col:=3;
                  y:=2;
                end;
             end;
end;
end;

{MUDA A DRIVE ACTUAL}
procedure mudar_drive;
begin
  clrscr;
  textbackground(white);
  textcolor(black);
  gotoxy(34,11);write('┌────────────┐');
  gotoxy(34,12);write('│');
  gotoxy(46,12);write('│');
  gotoxy(34,13);write('└────────────┘');
  textcolor(white);
  gotoxy(36,12);write('Drive ?');
  repeat
    gotoxy(43,12);write('  ');
    gotoxy(43,12);
    buflen:=2;
    read(drive);
 until ((drive='A:')or(drive='B:'));
 chdir(drive);
end;

{COLOCA O CURSOR NA ULTIMA LETRA DO TEXTO}
procedure fim_de_texto;
begin
  x:=max_linha_escrita;
  y:=max_col_escrita;
  linha:=22;
  col:=(max_col_escrita+1);
end;

{COLOCA O CURSOR NA PRIMEIRA LETRA DO TEXTO}
procedure comeco_de_texto;
begin
  x:=min_linha_escrita;
  y:=min_col_escrita;
  linha:=2;
  col:=(min_col_escrita+1)
end;

{LISTA O CONTEUDO DE PARTE DA TABELA NO ECRAN}
procedure listar_ecran;
begin
  d:=x-(linha-2);
  for c:=2 to 22 do
   begin
     gotoxy(2,c);write(lista[d]);
     d:=d+1;
   end;
end;

{CARREGA O FICHEIRO COM O CONTEUDO DA LISTA}
procedure lista_ficheiro;
begin
  for c:=1 to max_linha_escrita do
   begin
     for d:=1 to maxcol do
       begin
         carac:=lista[c,d];
         write(arq_texto,carac);
         if (d=78) then write(arq_texto,RETURN);
       end;
   end;
end;

{CARREGA A TABELA COM O CONTEUDO DO FICHEIRO}
procedure ficheiro_lista;
begin
  anular;
  d:=1;
  while not eof(arq_texto) do
    begin
      readln(arq_texto,st);
      for c:=1 to maxcol do lista[d,c]:=st[c];
      d:=d+1;
    end;
end;

{ESCRITA DO QUADRO PRINCIPAL COM OS ELEMENTOS CONSTITUINTES}
procedure escrita_total;
begin
  quadro_principal;
  nome_do_arquivo;
  posicoes;
  listar_escrita;
  listar_ecran;
end;

{INTRUCOES SOBRE AS TECLAS A USAR NO EDITOR}
procedure ajuda;
var ficheiro       :text;
    tabela         :array[1..60,1..78]of char;
    minimo,maximo  :integer;
begin
  clrscr;
  {$I-}
  assign(ficheiro,'AJUDA.DAT');{FICHEIRO DE AJUDA}
  reset(ficheiro);
  {$I+}
  if ioresult<>0 then begin
                        textbackground(white);
                        textcolor(white);
                        write(^g);
                        gotoxy(30,11);write('O ficheiro AJUDA.DAT');
                        gotoxy(30,12);write('nao foi encontrado.');
                        textcolor(black);
                        janela('┌','┐','└','┘','─','│',28,52,10,13);
                        repeat until keypressed;
                        textcolor(white);
                        exit;
                      end;
  d:=1;
  while not eof(ficheiro) do
    begin
      readln(ficheiro,st);
      for c:=1 to 78 do tabela[d,c]:=st[c];
      d:=d+1;
    end;
  textbackground(white);
  textcolor(black);
  janela('┌','┐','└','┘','─','│',1,80,1,23);
  gotoxy(35,1);write('  Ajuda  ');
  textcolor(white);
gotoxy(14,24);
write('ESC -          ',chr(24),' -          ',chr(25),' -          PgUp-PgDn -
textcolor(brown);
gotoxy(20,24);write('Sair');gotoxy(30,24);write('Cima');
gotoxy(40,24);write('Baixo');gotoxy(60,24);write('Pagina');
textcolor(white);
tabela[4,2]:=chr(27);tabela[5,2]:=chr(26);tabela[6,2]:=chr(24);
tabela[7,2]:=chr(25);tabela[24,2]:=chr(27);tabela[26,6]:=chr(26);
window(2,2,79,22);
for c:=1 to 20 do
 begin
   gotoxy(1,c);write(tabela[c]);
 end;
maximo:=21;
minimo:=1;
repeat
  repeat
    read(kbd,opcao);
  until opcao=#27;
  if keypressed then begin
                       read(kbd,opcao);
                       case opcao of
                                   'H':begin
                                         if minimo>1 then
                                          begin
                                            minimo:=minimo-1;
                                            maximo:=maximo-1;
                                            gotoxy(1,1);insline;
                                            gotoxy(1,1);write(tabela[minimo]);
                                            gotoxy(1,1);
                                          end;
                                       end;
                                   'P':begin
                                         if maximo<60 then
                                          begin
                                            minimo:=minimo+1;
                                            maximo:=maximo+1;
                                            gotoxy(1,1);delline;
                                            window(1,1,80,25);
                                            gotoxy(2,22);write(tabela[maximo]);
                                            window(2,2,79,22);
                                            gotoxy(1,20);
                                          end;
                                       end;
                                   'Q':begin
                                         clrscr;
                                         if maximo<41 then minimo:=minimo+20;
                                         if maximo>40 then begin
                                                             maximo:=40;
                                                             minimo:=40;
                                                           end;
                                         window(1,1,80,25);
                                         for c:=2 to 22 do
                                          begin
                                            gotoxy(2,c);write(tabela[maximo]);
                                            maximo:=maximo+1;
                                          end;
                                         maximo:=maximo-1;
                                         window(2,2,79,22);
                                         gotoxy(1,20);
                                       end;
                                   'I':begin
                                         clrscr;
                                         if minimo>20 then maximo:=maximo-20;
                                         if minimo<21 then begin
                                                             minimo:=21;
                                                             maximo:=21;
                                                           end;
                                         window(1,1,80,25);
                                         for c:=22 downto 2 do
                                           begin
                                             gotoxy(2,c);write(tabela[minimo]);
                                             minimo:=minimo-1;
                                           end;
                                         minimo:=minimo+1;
                                         window(2,2,79,22);
                                         gotoxy(1,1);
                                        end;
                               end:{ case }
                     end; { If }
  until opcao=#27;
  window(1,1,80,25);
end;

{MODIFICA O NOME DO FICHEIRO ACTUAL}
procedure mudar_nome;
begin
  quadro_2;
  fich2:=fich1;
  escrita_total;
end;

{GRAVA O FICHEIRO ACTUAL}
procedure gravar;
begin
  clrscr;
  repeat
    quadro_1;
    esperar;
    {$I-}
    assign(arq_texto,fich1);
    rewrite(arq_texto);
    lista_ficheiro;
    pass:=true;
    close(arq_texto);
    {$I+}
    if ioresult<>0 then begin
                          quadro_3;
                          if opcão=#27 then begin
                                              escrita_total;
                                              pass:=false;
                                              exit;
                                            end;
                          pass:=false;
                        end;
  until pass=true;
end;

procedure gravar_ficheiro;
begin
  gravar;
  if pass=false then exit;
  gravacao:=true;
  caixa;
  gotoxy(50,12);write('gravado');
  repeat until keypressed;
  escrita_total;
end;

{CONFIRMA SE O FICHEIRO ACTUAL FOI GRAVADO}
procedure confirmar_gravacao;
begin
  if gravacao=false then
   begin
     quadro_4;
     if opcão in['S','s'] then
      begin
        gravar;
        caixa;
        gotoxy(50,12);write('gravado');
        repeat until keypressed;
      end;
     clrscr;
   end;
end;

{CRIA UM FICHEIRO}
procedure criar_ficheiro;
begin
```

```
   quadro_2;
   pass:=true;
   gravacao:=false;
   fich2:=fich1;
   caixa;
   gotoxy(50,12);write('criado');
   repeat until keypressed;
   clrscr;
end;

{LE UM FICHEIRO DA DISKETE}
procedure ler_ficheiro;
begin
 repeat
   quadro_2;
   quadro_1;
   esperar;
   pass:=true;
   {$I-}
   assign(arq_texto,fich1);
   reset(arq_texto);
   {$I+}
   if ioresult<>0 then begin
                        quadro_3;
                        pass:=false;
                        if opcao=#27 then begin
                                           fich1:=fich2;
                                           exit;
                                          end;
                       end;
 until pass=true;
 anular;
 ficheiro_lista;
 close(arq_texto);
 gravacao:=false;
 fich2:=fich1;
 caixa;
 gotoxy(50,12);write('carregado');
 repeat until keypressed;
 clrscr;
end;

{PROCEDIMENTO PRINCIPAL}
procedure entrada;
var passagem:boolean;
begin
  col:=2;
  linha:=2;
  x:=1;
  y:=1;
  gravacao:=false;
  tipo_de_escrita:=true;
  getdir(0,drive);
  escrita_total;
  repeat
    repeat
      posicoes;
      gotoxy(col,linha);
      passagem:=true;
      read(kbd,carac);
      if (ord(carac)=8)then begin
                             apagar_tras;
                             passagem:=false;
                            end;
      if (ord(carac)=7) then begin
                              apagar_frente;
                              passagem:=false;
                             end;
      if (ord(carac)=9) then begin
                              tabulador;
                              passagem:=false;
                             end;
      if (carac=^Q)then begin
                         apagar_atras_cursor;
                         passagem:=false;
                        end;
      if (carac=^E)then begin
                         apagar_frente_cursor;
                         passagem:=false;
                        end;
      if(carac=^Z)then begin
                        palavra_atras;
                        passagem:=false;
                       end;
      if(carac=^X)then begin
                        palavra_frente;
                        passagem:=false;
                       end;
      if(carac=^P)then begin
                        mudar_drive;
                        escrita_total;
                        passagem:=false;
                       end;
      if(carac=^D)then begin
                        fim_de_texto;
                        escrita_total;
                        passagem:=false;
                       end;
      if(carac=^A)then begin
                        comeco_de_texto;
                        escrita_total;
                        passagem:=false;
                       end;
      if(carac=^O)then begin
                        mudar_nome;
                        passagem:=false;
                       end;
    until passagem=true;
    functionkey:=false;
    if ((carac=#27)and keypressed)or(carac=RETURN) then
    begin
      if carac<>RETURN then begin
                              read(kbd,carac);
                              functionkey:=true;
                             end
                             else functionkey:=true;
      case carac of
             'H':cursor_cima;
      RETURN,'P':cursor_baixo;
             'M':cursor_direita;
             'K':cursor_esquerda;
             'I':scroll_pagina(up);
             'Q':scroll_pagina(down);
             'A':retirar_linha;
             'B':inserir_linha;
             'G':cursor_principio;
             'O':cursor_fim;
             'R':begin
                   tipo_de_escrita:=not (tipo_de_escrita);
                   listar_escrita;
                 end;
             ';':begin
                   ajuda;
                   escrita_total;
                 end;
             '<':begin
                   tabela_ascii;
                   escrita_total;
                 end;
             '=':begin
                   confirmar_gravacao;
                   ler_ficheiro;
                   escrita_total;
                 end;
             '>':begin
                   confirmar_gravacao;
                   esperar;
                   anular;
                   criar_ficheiro;
                   escrita_total;
                 end;
             '?':gravar_ficheiro;
             'C':begin
                   substituir;
                   escrita_total;
                 end;
             'D':begin
                   mudar_de_linha(x);
                   escrita_total;
                 end;
          end; { Case }
    end; { If }
    if (functionkey=false)and(carac<>#27) then
     begin
       if tipo_de_escrita=True then escrita_inserida
        else
          begin
            lista[x,y]:=carac;
            gotoxy(col,linha);write(carac);
          end;
       cursor_direita;{AVANCA UMA POSICAO}
     end;
  until carac=#27;
end;

procedure editor;
begin
  clrscr;
  anular;
  apresentacao;
  repeat
    quadro;
    case opcao of
          'l','L':ler_ficheiro;
          'c','C':criar_ficheiro;
          's','S':exit;
         end;
  until pass=true;
  repeat
    entrada;
    clrscr;
    textbackground(white);
    textcolor(black);
    janela('┌','┐','└','┘','─','│',28,52,11,13);
    textcolor(white);
    gotoxy(30,12);write('Tem a certeza (S/N)?');
    repeat
      read(kbd,opcao);
    until opcao in['s','n','N','S'];
  until opcao in['s','S'];
  confirmar_gravacao;
  clrscr;
end;


begin
  editor;
end.
```

2.º PRÉMIO

MANUEL JOSÉ RASQUILHO
LINDA-A-VELHA

É um jogo interessante, com o defeito de já ser muito conhecido, no entanto tem algumas diferenças em relação a outras versões. O programa em si tem qualidade gráfica, embora como sabemos, em Turbo Pascal é só chamar as rotinas, logo a dificuldade não é grande.

Em termos de estruturas do programa, está bom e tem qualidade.

A opção de jogo a dois tem interesse e é ao mesmo tempo criativa.

```
Program Enforcado;

Type
     comp_palav=string[21];
     comp_mensag=string[20];

Const
      max_dados=1000;    { Maximo de 1000 palavras no ficheiro }

Var
   escolha:char;
   varios:integer;
   intfile:file of comp_palav;

{$I GRAPH.F}

Procedure apresentacao;

Var x:integer;

Begin
     clrscr;
     graphcolormode;
     palette(2);
     For x:=1 to 80 do
         begin
              circle(x,10,10,1);
              x:=x+3;
              circle(309,x,10,2);
              delay(10);
         end;
     For x:=10 to 180 do
         Begin
              circle(x,190,10,1);
              x:=x+3;
              circle(10,x,10,2);
              delay(15);
         end;
     For x:=25 to 114 do
         begin
              draw(25,26,x,175,1);
              x:=x+2;
         end;
     For x:=26 to 115 do
         begin
              draw(293,175,x,25,2);
              x:=x+2;
         end;
     For x:=35 to 158 do
         begin
              delay(10);
              draw(111,x,216,x,1);
         end;
     gotoxy(16,6);
     writeln('             ');
     gotoxy(16,7);
     writeln(' RASQUILHO   ');

     gotoxy(16,8);
     writeln(' SOFTWARE    ');
     gotoxy(16,9);
     writeln(' PRESENTS    ');
     gotoxy(16,10);
     writeln('             ');
     gotoxy(16,11);
     writeln('             ');
     gotoxy(16,12);
     writeln(' ENFORCADO   ');
     gotoxy(16,13);
     writeln('             ');
     gotoxy(16,14);
     writeln('             ');
     gotoxy(16,15);
     writeln('    por      ');
     gotoxy(16,16);
     writeln('             ');
     gotoxy(16,17);
     writeln('  Manuel     ');
     gotoxy(16,18);
     writeln(' Rasquilho   ');
     gotoxy(16,19);
     writeln('             ');
     repeat until keypressed;

end;


Procedure som(numero,demora,amp:integer);

var
   cons:real;
   freq,loop:integer;

begin
      cons:=exp(ln(2)/12);
      freq:=round(amp*exp(ln(cons)*numero));
      for loop:=trunc(freq/2) to freq do
                sound(loop);
      delay(demora);
      nosound
end;


Function existe_ficheiro:boolean;


begin
     assign(intfile,'TITULOS');
     {$I-} reset(intfile) {$I+};
     existe_ficheiro:=(ioresult=0);

end;


Function maiuscula(nome:comp_palav):comp_palav;

var
     caracter    :char;
     ord_caract  :integer;

    { Converter a primeira letra em maiuscula. }

begin
     caracter:=copy(nome,1,1);
     ord_caract:=ord(caracter);
     if (ord_caract>96) and (ord_caract<123) then
        begin
             ord_caract:=ord_caract-32;
             caracter:=chr(ord_caract);
             delete(nome,1,1);
             insert(caracter,nome,1);
        end;
     maiuscula:=nome;

end;


Procedure ficheiro;

var
   nome_st    :comp_palav;
   comp_nome  :integer;


Function fic_cheio:boolean;

begin
     assign(intfile,'TITULOS');
     reset(intfile);
     if filesize(intfile)>=max_dados then fic_cheio:=true
                                     else fic_cheio:=false;
     close(intfile);

end;


begin
     som(1,100,200);
     repeat
           repeat
                 clrscr;
                 Graphcolormode;
                 palette(2);
                 if existe_ficheiro=false then
                    begin
                         assign(intfile,'TITULOS');
                         rewrite(intfile);
                         close(intfile);
                    end;
                 if fic_cheio=true then
                    begin
                         som(10,100,200);
                         writeln('ATENCAO : Ficheiro de dados esta');
                         writeln('           cheio.');
                         repeat until keypressed;
                         exit;

                end;
             textcolor(2);
             gotoxy(5,23);
             write('Nao utilize acentos nem cedilhas.');
             textcolor(3);
             gotoxy(10,25);
             write('Prima <Enter> para sair.');
             gotoxy(1,1);
             write('Insira a palavra -->');
             read(nome_st);
             comp_nome:=length(nome_st);
             if comp_nome>20 then
                begin
                     clrscr;
                     graphcolormode;
                     palette(2);
                     som(10,100,200);
                     writeln('ATENCAO : So se pode utilizar');
                     writeln('            palavras ate 20 letras.');
                     repeat until keypressed;
                end;
          until comp_nome<=20;
          if nome_st<>'' then
             begin
                  nome_st:=maiuscula(nome_st);
                  assign(intfile,'TITULOS');
                  reset(intfile);
                  seek(intfile,filesize(intfile));
                  write(intfile,nome_st);
                  close(intfile);
             end;
     until nome_st='';

end;


Procedure escolher(var varios:integer);

var
   posicao_y,demora,numero,amp,pos,compalavra,tipo_car,letras_certas,
   posicao_x,ang,ang_div,posicao_car,comp,pos_letra,f,hipotese,fim,loc:integer;
   palavra:comp_palav;
   prim_letra,letra,car:char;
   cad_pos:array[1..15] of integer;
   cad_total:array[1..15] of char;
   mensagem:comp_mensag;
   car_correcto,vencedor,repeticao,term:boolean;


Procedure rand(var varios:integer);

var
   ran,tamanho:integer;

begin
     If varios=0 then
        begin
             assign(intfile,'titulos');
             reset(intfile);
             tamanho:=filesize(intfile);
```

```
            ran:=random(tamanho);
            if ran=tamanho then ran:=ran-1;
            seek(intfile,ran);
            read(intfile,palavra);
            close(intfile);
        end
      else
        begin
          repeat
            clrscr;
            Graphcolormode;
            palette(2);
            write('Qual e a Palavra -->');
            read(palavra);
            if length(palavra)>20 then
               begin
                     clrscr;
                     graphcolormode;
                     palette(2);
                     som(10,100,200);
                     writeln('ATENCAO : So se pode utilizar');
                     writeln('             palavras ate 20 letras.');
                     repeat until keypressed;
               end;
          until length(palavra)<=20;
            palavra:=maiuscula(palavra);
        end;

end;


Begin
     if (existe_ficheiro=false) and (varios=0)then
        begin
             clrscr;
             graphcolormode;
             palette(2);
             som(10,100,200);
             writeln('ATENCAO : Nao existe ficheiro com');
             writeln('          palavras.Primeiro tem que ser');
             writeln('          criado.');
             repeat until keypressed;
             exit;
        end;
     rand(varios);
     clrscr;
     graphcolormode;
     palette(2);
     draw(0,0,319,0,1);
     draw(319,0,319,199,1);
     draw(0,199,319,199,1);
     draw(0,0,0,199,1);
     fillpattern(280,15,285,185,1);
     fillpattern(230,15,280,20,1);
     fillpattern(265,186,300,195,1);
     hipotese:=7;term:=false;fim:=0;vencedor:=false;pos:=0;
     comp:=length(palavra);
     prim_letra:=copy(palavra,1,1);
     gotoxy(7,4);
     write(prim_letra);

for f:=1 to (comp-1) do
    begin
         gotoxy(7+f,4);
         write('- ');
    end;
for f:=1 to 20 do cad_total[f]:=' ';
repeat
      car_correcto:=false;letras_certas:=0;
      textcolor(2);
      Gotoxy(2,12);
      write('Introduza a letra');
      textcolor(3);
      gotoxy(2,21);
      writeln('Letras usadas: ');
     repeat
      read(kbd,car);
      tipo_car:=ord(car);
     until (tipo_car>96) and (tipo_car<123);
      For pos_letra:=2 to comp do
          begin
               letra:=copy(palavra,pos_letra,1);
               if car=letra then
                                begin
                                  letras_certas:=letras_certas+1;
                                  cad_pos[letras_certas]:=(pos_letra-1);
                                  car_correcto:=true;
                                  fim:=fim+1;
                                end;
          end;
      loc:=0;repeticao:=false;
      for f:=1 to 20 do if cad_total[f]=car then
                            begin
                                 fim:=fim-letras_certas;
                                 repeticao:=true;
                            end;
      if repeticao=false then
         begin
              for f:=1 to 20 do if (cad_total[f])<>' ' then loc:=loc+1;
              cad_total[loc+1]:=car;
              if car_correcto=true then
                 begin
                    { Escrever no ecrä as letras iguais }
                     for f:=1 to letras_certas do
                                      begin
                                           posicao_car:=cad_pos[f];
                                           gotoxy(7+posicao_car,4);
                                           write(car);
                                      end;
                 end
               else
                 begin
                    { Se näo existe a letra }
                      hipotese:=hipotese-1;
                      gotoxy(16+pos,21);
                      write(car);
                      pos:=pos+1;
                      case hipotese of
                                 6:circle(235,31,10,1);
                                 5:draw(235,41,235,110,1);
                                 4:draw(235,55,220,85,1);
                                 3:draw(235,55,250,85,1);
                                 2:draw(235,110,210,135,1);
                                 1:draw(235,110,260,135,1);
                            end;
                   end;
               If hipotese=1 then term:=true;
               if fim=(cómp-1) then
                  begin
                       term:=true;
                       vencedor:=true;
                  end;
         end;
Until term=true;
if vencedor=true then
   Begin
        circle(235,31,10,0);
        draw(235,41,235,110,0);
        draw(235,55,220,85,0);
        draw(235,55,250,85,0);
        draw(235,110,210,135,0);
        {Começo em baixo}
        draw(220,170,195,195,1);
        draw(220,170,245,195,1);
        draw(221,115,235,145,1);
        draw(219,115,205,145,1);
        draw(220,101,220,170,1);
        circle(220,91,10,1);
        {Levantar os braços}
        for ang:=0 to 70 do
            Begin
                 ang_div:=abs(ang div 4);
                 draw(221,115,235+ang_div,145-ang,1);
                 draw(219,115,205-ang_div,145-ang,1);
                 delay(20);
                 draw(221,115,235+ang_div,145-ang,0);
                 draw(219,115,205-ang_div,145-ang,0);
            end;
        draw(221,115,252,75,1);
        draw(219,115,188,75,1);
        {Musica de Victoria}
        som(7,100,100);
        som(7,100,100);
        som(9,250,100);
        som(7,250,100);
        som(12,250,100);
        som(11,300,100);
        som(7,100,100);
        som(7,100,100);
        som(9,250,100);
        som(7,250,100);
        som(14,250,100);
        som(12,300,100);
        som(7,100,100);
        som(7,100,100);
        som(19,250,100);
        som(16,250,100);
        som(12,250,100);
        som(11,250,100);
        som(9,300,100);
        som(17,100,100);

        som(17,100,100);
        som(16,250,100);
        som(12,250,100);
        som(14,250,100);
        som(12,300,100);
        mensagem:='P A R A B E N S';
        compalavra:=22;
        gotoxy(2,21);write('
   end
  else
   begin
      { Esticar as pernas }
        for ang:=1 to 20 do
            begin
                 draw(235,110,209+ang,134+ang,0);
                 draw(235,110,261-ang,134+ang,0);
                 draw(235,110,210+ang,135+ang,1);
                 draw(235,110,260-ang,135+ang,1);
                 delay(100);
            end;
        delay(1500);
        { Apagar o morto }
        circle(235,31,10,0);
        draw(235,41,235,110,0);
        draw(235,55,220,85,0);
        draw(235,55,250,85,0);
        draw(235,110,230,155,0);
        draw(235,110,240,155,0);
        { Fazer a campa }
        arc(210,165,180,15,1);
        draw(210,165,240,165,1);
        fillshape(215,162,1,1);
        fillpattern(210,165,240,195,1);
        { Fazer Cruz }
        fillpattern(224,165,226,190,0);
        fillpattern(219,170,231,172,0);
        {Musica de derrota}
        som(9,600,100);
        som(9,600,100);
        som(8,250,100);
        som(9,600,100);
        som(12,600,100);
        som(11,250,100);
        som(11,600,100);
        som(9,250,100);
        som(9,600,100);
        som(8,250,100);
        som(9,600,100);
        mensagem:='P E R D E S T E';
        if length(palavra)<=10 then
           begin
                posicao_x:=16;
                posicao_y:=21;
                compalavra:=length(palavra)+16;
           end
          else
           begin
                posicao_x:=4;
                posicao_y:=22;
                compalavra:=22;
```

```
            end;
          gotoxy(2,21);
          write('A palavra era              ');
          gotoxy(posicao_x,posicao_y);
          write(palavra);
        end;
     repeat
        gotoxy((round((compalavra)/2))-7,19);
        write(mensagem);
        delay(300);
        gotoxy((round((compalavra)/2))-7,19);
        write('                    ');
        delay(300);
     until keypressed;


end;

begin
     apresentacao;
     som(1,100,200);
     repeat
         clrscr;
         graphcolormode;
         palette(2);
         draw(0,0,319,0,1);
         draw(319,0,319,199,1);
         draw(0,199,319,199,1);
         draw(0,0,0,199,1);
         draw(0,24,319,24,1);
         gotoxy(12,2);
         write('E N F O R C A D O');
         gotoxy(13,23);
         write('Esc - Para sair');
         textcolor(1);
         gotoxy(3,7);
         writeln('Escolha : ');
         textcolor(3);
         gotoxy(5,9);
         write('1- Acrescentar o Vocabulario.');
         gotoxy(5,11);
         write('2- Jogar com o Computador.');
         gotoxy(5,12);
         write('3- Jogar com dois Jogadores.');
         read(kbd,escolha);
         case escolha of
                         '1':ficheiro;
                     '2','3':begin
                               som(1,100,200);
                               if escolha='2' then varios:=0
                                              else varios:=1;
                               escolher(varios);
                            end;
             end;
     until (escolha=chr(27)) and not keypressed;
     clrscr;

end.
```

3.º PRÉMIO

ALEXANDRE MANUEL TAVARES BRAGANÇA

OLIVEIRA DE AZEMÉIS

Tem boa apresentação inicial, com cores e efeitos engraçados.

É um jogo simples, mas que requer uma certa atenção e agilidade, se jogado *sem* o uso da tecla «F1». Neste caso se encostar-mos o buraco que apanha as cubas ao lado esquerdo e premir-mos consecutivamente a tecla «F1», nunca mais se «perde» e é só somar os pontos, o que acho ser a única «fraqueza» do programa.

É um jogo que requer pouco trabalho, no entanto pela listagem apresentada foi muito bem estruturado e revela bons conhecimentos de Turbo-Pascal. Como o próprio criador o indica se modificar o procedimento «sair peças», talvez consiga aumentar a necessidade de perícia para o jogo.

Não encontrei qualquer erro.

```
  Programa de Alexandre Braganca - 1989 }

program square;
uses crt,dos,graph;
type
  tipo_ecra=array[1..17,1..7] of byte;
  p_cair=^cair;
  cair=record
        a,b:byte;
        prox,ant:p_cair;
      end;

var
  ecra:tipo_ecra;
  registos:registers;
  colunas:array[1..17] of byte;
  i_cair,f_cair:p_cair;
  colector0,colector:byte;
  p_colectadas,max:real;
  fim_de_jogo,sair,sai:boolean;
  car:char;
  n_1000:longint;
  saidas,saidas0:array[1..9] of byte;
  temp2,temp20:p_cair;
  va,vb:word;
  placa,modo:integer;

Procedure cursor_off;
Begin
  registos.ah:=1; registos.ch:=$20; registos.cl:=$20; Intr(16,registos);
End;

Procedure cursor_on;
Begin
  registos.ah:=1; registos.ch:=$6; registos.cl:=$7; Intr(16,registos);
End;

procedure esc_jogo;
var
  str1,str2,str3,str4:string;
  a:byte;
begin
  window(1,1,40,25);
  textcolor(white);
  textbackground(black);
  clrscr;
  str1:='╔══════════════════════════╗';
  str2:='║                          ║';
  str3:='╚══════════════════════════╝';
  gotoxy(1,1); write(str1,str2,str3);
  gotoxy(3,2); write(' SquareGame');
  textcolor(white);
  textbackground(black);
  gotoxy(21,2);
  write('R');
  textbackground(black);
  write('(      ) ');
  textbackground(white);
```

```
  write(' ');
  textbackground(black);
  write('(     )');
  gotoxy(1,4);
  write(str1);
  str2:=' ║╔═╗╔═╗╔═╗╔═╗╔═╗╔═╗╔═╗╔═╗╔═╗║ ';
  str3:=' ║║ ║║ ║║ ║║ ║║ ║║ ║║ ║║ ║║ ║║ ';
  str4:=' ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ╚╝ ';
  write(str2,str3,str4);
  str2:=' ╔╗                          ╔╗ ';
  str3:=' ║║                          ║║ ';
  str4:=' ╚╝                          ╚╝ ';
  write(str2);
  for a:=1 to 12 do write(str3);
  write(str4);
  str1:=' ╔════════════════════════════╗ ';
  str2:=' ║                            ║ ';
  str3:=' ╚════════════════════════════╝';
  write(str1,str2,str2,str3);
end;

procedure esc_peca(tipo:byte;x,y:integer);
{    se tipo = 0 -> peca preta (fundo)
     se tipo = 1 -> peca branca   }
begin
  case tipo of
    0:begin textbackground(black); textcolor(black); end;
    1:begin textbackground(white); textcolor(white); end;
  end;
  gotoxy(x,y);
  write('  ');
  gotoxy(x,y+1);
  write('  ');
  textbackground(black);
end;

procedure esc_pontos;
begin
  textbackground(black);
  textcolor(white);
  gotoxy(32,2);
  write(p_colectadas:5:0);
  if (p_colectadas>=max) then
  begin
    max:=p_colectadas;
    gotoxy(23,2);
    write(max:5:0);
  end;
end;

procedure f1;
var
  a:byte;
  temp:p_cair;
  n:byte;
begin
  n:=colunas[1];
  for a:=1 to 16 do
    colunas[a]:=colunas[a+1];
  colunas[17]:=n;
  temp:=i_cair;
  while (temp<>nil) do
  begin

procedure esc_ecra;
var
  a,b:byte;
begin
  for a:=1 to 9 do esc_peca(saidas[a],4+((a-1)*4),6);
  for a:=1 to 17 do
    for b:=1 to 7 do
      esc_peca(ecra[a,b],4+(a-1)*2,8+(b-1)*2);


end;

procedure entrou_peca(cor:byte);
var
  a:byte;
begin
  for a:=0 to 5 do
  begin
    textbackground(cor);
    sound(a*100);
    gotoxy(4+((colector-1)*2),22);
    write('  ');
    gotoxy(4+((colector-1)*2),23);
    write('  ');
    delay(2);
    textbackground(black);
    gotoxy(4+((colector-1)*2),22);
    write('  ');
    gotoxy(4+((colector-1)*2),23);
    write('  ');
    nosound;
  end;
  esc_pontos;
end;

procedure colectar;
begin
  if colunas[colector]>0 then
  begin
    colunas[colector]:=colunas[colector]-1;
    p_colectadas:=p_colectadas+10;
    construir_ecra;
    esc_ecra;
    entrou_peca(white);
  end;
end;

procedure esc_colector;
begin
  textcolor(white);
  textbackground(black);
  gotoxy(3+((colector0-1)*2),22);
  write('════');
  gotoxy(3+((colector0-1)*2),23);
  write('    ');
  gotoxy(3+((colector0-1)*2),24);
  write('    ');
  gotoxy(3+((colector-1)*2),22);
  write('╖  ╓');
  gotoxy(3+((colector-1)*2),23);
  write('║  ║');
  gotoxy(3+((colector-1)*2),24);
  write('╚══╝');
end;
```

```
    if temp^.a=1 then temp^.a:=17
    else temp^.a:=temp^.a-1;
    temp:=temp^.prox;
  end;
end;

procedure cair_pecas;
var
  pont:p_cair;
begin
  pont:=i_cair;
  while pont<>nil do
  begin
    pont^.b:=pont^.b+1;
    pont:=pont^.prox;
  end;
end;

procedure entra_coluna;
var
  pont,temp1:p_cair;
  col,lin:byte;
begin
  pont:=i_cair;
  while pont<>nil do
  begin
    col:=pont^.a;
    lin:=pont^.b;
    if (lin>=(7-colunas[col])) then
    begin
      temp1:=pont;
      pont:=pont^.prox;
      if temp1^.ant<>nil then temp1^.ant^.prox:=pont
      else i_cair:=pont;
      if pont<>nil then pont^.ant:=temp1^.ant
      else f_cair:=temp1^.ant;
      dispose(temp1);
      colunas[col]:=colunas[col]+1;
    end
    else pont:=pont^.prox;
  end;
end;

procedure coluna_cheia;
var
  a:byte;
begin
  for a:=1 to 17 do
    if colunas[a]>=7 then fim_de_jogo:=true;
end;

procedure sair_pecas;
var
  a:word;
  lin,col:byte;
  temp:p_cair;
begin
  for a:=1 to 9 do saidas[a]:=0;
  lin:=0;
  col:=random(9999)+1;
  if (col<((n_1000+1)*100)) then
  begin
    a:=0;
    repeat
      col:=random(18);
      if (col in [1,3,5,7,9,11,13,15,17]) and
         (saidas[(col div 2)+(col mod 2)]=0) and
         (saidas0[(col div 2)+(col mod 2)]=0) then
      begin
        new(temp);
        temp^.a:=col;
        temp^.b:=lin;
        temp^.prox:=nil;
        temp^.ant:=nil;
        if i_cair=nil then
        begin
          i_cair:=temp;
          f_cair:=temp;
        end
        else
        begin
          temp^.prox:=i_cair;
          i_cair^.ant:=temp;
          i_cair:=temp;
        end;
        saidas[(col div 2)+(col mod 2)]:=1;
      end;
      a:=a+10;
    until a>n_1000;
  end;
  saidas0:=saidas;
end;

procedure actual_n_1000;
begin
  n_1000:=trunc(p_colectadas/1000);
end;

procedure construir_ecra;
var
  temp1:p_cair;
  col:byte;
  a,b:byte;
begin
  fillchar(ecra,119,chr(0));
  temp1:=i_cair;
  while temp1<>nil do
  begin
    if temp1^.b<>0 then ecra[temp1^.a,temp1^.b]:=1;
    temp1:=temp1^.prox;
  end;
  for a:=1 to 17 do
  begin
    if (colunas[a]>0) then
    begin
      b:=0;
      col:=7;
      while (b<>colunas[a]) do
      begin
        ecra[a,col]:=1;
        col:=col-1;
        b:=b+1;
      end;
    end;
  end;
end;
```

```
procedure inicia_var;
var
  a:byte;
begin
  fillchar(ecra,119,chr(0));
  fim_de_jogo:=false;
  p_colectadas:=0;
  i_cair:=nil;
  f_cair:=nil;
  for a:=1 to 17 do colunas[a]:=0;
  for a:=1 to 9 do saidas0[a]:=0;
  colector0:=9;
  colector:=9;
end;

begin
  sair:=false;
  max:=0;
  repeat
    placa:=cga;
    modo:=cgac0;
    initgraph(placa,modo,'');
    settextstyle(2,0,4);
    outtextxy(220,186,'(c) Alex, 1989');
    settextstyle(3,0,6);
    repeat
      va:=1;
      repeat
        setcolor(va);
        outtextxy(8,40,'SquareGame');
        line(8,93,308,93);
        va:=va+1;
      until (va=4) or (keypressed);
    until keypressed;
    car:=readkey;
    if ord(car)=27 then sair:=true
    else
    begin
      closegraph;
      textmode(co40);
      inicia_var;
      cursor_off;
      esc_jogo;
      esc_colector;
      gotoxy(23,2);
      write(max:5:0);
      esc_pontos;
      repeat
        actual_n_1000;
        colectar;
        cair_pecas;
        sair_pecas;
        entra_coluna;
        construir_ecra;
        esc_ecra;
        coluna_cheia;
        if keypressed then
        begin
          car:=readkey;
          if ord(car)=0 then
          begin
            car:=readkey;
            case ord(car) of
              75:begin
                   colector0:=colector;
                   if colector>1 then colector:=colector-1
                   else colector:=17;
                   esc_colector;
                 end;
              77:begin
                   colector0:=colector;
                   if colector<17 then colector:=colector+1
                   else colector:=1;
                   esc_colector;
                 end;
              59:f1;
            end;
          end;
        end;
      until (fim_de_jogo);
      sai:=false;
      va:=1;
      construir_ecra;
      esc_ecra;
      delay(1000);
      repeat
        if keypressed then car:=readkey;
      until not(keypressed);
      repeat
        if colunas[va]>=7 then
        begin
          for vb:=20 downto 0 do
          begin
            sound(vb*10);
            delay(50);
            nosound;
          end;
          repeat
            for vb:=7 downto 1 do esc_peca(0,4+(va-1)*2,8+(vb-1)*2);
            delay(100);
            for vb:=7 downto 1 do esc_peca(1,4+(va-1)*2,8+(vb-1)*2);
            delay(200);
          until keypressed;
          car:=readkey;
          sai:=true;
        end
        else va:=va+1;
      until sai;
      temp2:=i_cair;
      while temp2<>nil do
      begin
        temp20:=temp2;
        temp2:=temp2^.prox;
        dispose(temp20);
      end;
    end;
  until sair;
  closegraph;
  textmode(co80);
  clrscr;
  cursor_on;
end.
```

COMPRO/TROCO/VENDO

COMPRO

Livro ou manual em Português de código de máq. 8086—Contactar: José Cruz—Quinta da Mata, L4B r/c Esq.
3080 FIGUEIRA DA FOZ

MSX—DOS—E—DRIVE 3/50—Para: MSX—Phillips
Telef. 049-67644 depois das 20 horas

Jogos e programas para o CPC 64K—Contactar: Rui Pedro Faria Ricardo-Rua Australias, 564-4.º Dt.º
4450 MATOSINHOS

AMSTRAD 1512 OU 1640 em 2.ª mão; preço económico—Contactar: Daniel António Ferreira Lemos—Rua do Caixa Santiago 3500 VISEU

Floppy disc para o AMSTRAD CPC 6128 Jogos e Programas Telef. 052—22531 ou 684622 Impressora p/ PC1 da OLIVETTI Formato A-4. Fernando M. L. Cruz-Rua Vasco da Gama, 12-1.º Dt.º
2735 RIO DE MOURO —CACÉM

Monitor NEPTUN ou PHILLIPS para Spectrum—Contactar: Rui Inácio; 049-312329; R. Marquês de Pombal, 18
2300 TOMAR

Jogos e Programas para ATARI 520STFM—Contactar: Nuno Lopes—telef. 27507 em Setúbal depois das 21.00

AMSTRAD PC 1512/1640 em 2.ª mão Policromático—Telefone 9958778.
LEÇA DA PALMEIRA

Jogos para o VG-802 MSX—Contactar depois das 19h: 062-35138

Desejo adquirir um computador AMSTRAD. É favor enviar preços. Carlos Augusto Garcias Valente—R. da Liberdade, 1.º bloco 2.º Dt.º—7860 MOURA

Jogos para RC em 3.5 a preços razoáveis. Enviar lista a: Rui André Gonçalves—Av. Porfírio da Silva, 53 r/c Dt.º
4700 BRAGA

Impressora para AMSTRAD PC até 50 cts dependendo do estado. Carlos P.—R. Mouzinho de Albuquerque, 10-2.º Esq.
Telef. 82651
8500 PORTIMÃO

TROCO

LOTUS 123 DBASE III por TURBOCAD ou AUTOCAD—Contactar: António Jorge Mesquita —Telef. 7110334
VILA NOVA DE GAIA

Quattro por MSX-DOS; Telef. 049-67644-depois das 20 horas. Variado tipo de Software p/ PC'S. Troco lista—Contactar: 2735 AGUALVA CACÉM

Todo o tipo de programas para PC. Enviem listas—Contactar: Paulo Farinha Alves—Av. Marginal n.º 160, S. Pedro do Estoril
2765 ESTORIL

Software PC1640. Rogério Gonçalves; R. D. José Carcomo Lobo, 17- 2.º Esq. LAVRADIO —Telefone: 2043336

Jogos CPC K7 ou Disk. Tenho cerca de 150 jogos novidades cont: Ricardo Santos, Lote 17 Cruz d'Areia—2400 LEIRIA—Telef. 044-26644

Todo tipo de Programas PC. Troco lista—Contactar: Fernando José, Apartado 150—2505 CALDAS DA RAINHA Codex—Telef. (062) 34924

Todo o tipo de Software. Enviar lista ou morada para: Nuno Pinheiro, Crestelo—6270 SEIA

Jogo Leisure Suit Larry10V Roger Rabbit por Testdrive II em disketes de 3/5.Enviar para: Nuno Pinheiro—Crestelo—6270 SEIA

Software para PC em diskettes 3 1/2''—Contactar: TECNOSEARCH—Carlos Alberto Antunes, Barreiros Cepões
3500 VISEU

IBM PS2 por um Commodore PC Amiga 2000—Contactar: Luís Eduardo—Telef. 9329568

Todo o tipo de Software PC. Envia lista para: Paulo Santos kaku—Pr. Simão Veiga, R. T1 CB 8E—ST.º Ant.º Cavaleiros—2670 LOURES—Tel. 9885462

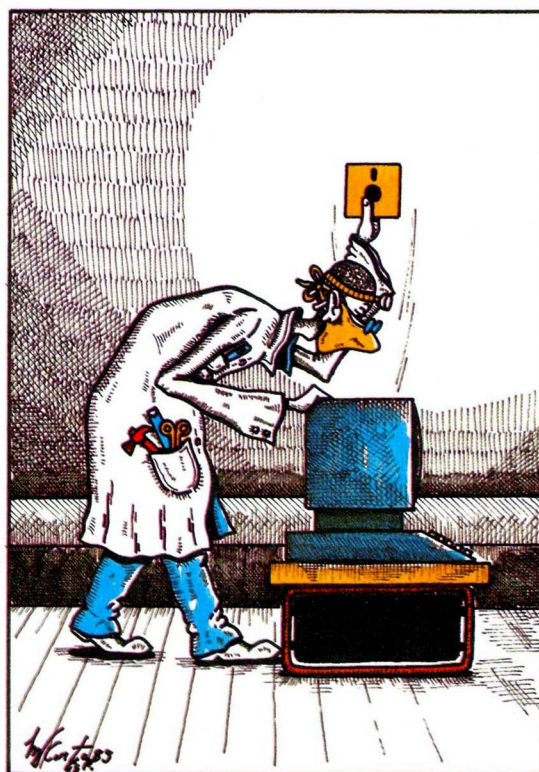
Soft para PC 5.25''. Envia lista e recebes a minha. Gabriela Magalhães—Rua Oscar Silva, 1741—2.º Dt.º, Leça da Palmeira—4450 MATOSINHOS

Todo o tipo de Jogos para o PC. Enviar lista para Vasco Miguel Marques, Lugar de Penouços Nogueira
4700 BRAGA—Telef. 973158

Software 5.25'', tenho Jogos, Aplicações. Procuro Compiladores C, CAD... Alexandre Viegas —manhã/tarde a partir das 6.00. Telefone: 613476
LISBOA

Flight Simulat. versão 3.00 por Pctools versão 4.21 ou outro programa em minha posse. Contactar: Rui, Av. Independência Nacional 4-6.º Esq.
2500 CALDAS DA RAINHA

Programas e/ou Jogos. Pascal, Basic e DBase são algumas das linguagens. Contactar—Ricardo Pinto; Travessa Fernandes Tomás—OVAR
Telefone: 53408.

Programas todo tipo. Lista com mais de 200 títulos. Envia a tua.Guilherme Teixeira. Rua Luciano Cordeiro, 14
5370 MIRANDELA

VENDO

AMSTRAD—1640-Mono-DD + HARDCARD-20m+rato com P-windows, facilito programas, 225 000$00.
Telefone: 9013346—PORTO

CASIO FX 7LOP, com 3 manuais em muito bom estado. Contactar: Telef. 563001 das 18h às 20h—PORTO
Preço: 100 000$00

Vasta gama de Software para IBM-PC, a baixo preço e garantido. Pedir lista para: Apartado 4361—1508 LISBOA Codex

Muitos jogos para PC. Enviem pedidos de lista e selo para resposta para: Zé. Baixo preço. Quinta da mata L4B r/c
3080 FIGUEIRA DA FOZ

AMSTRAD 1512- Mono, c/ 640 KRAM e HD de 64MB, por 225 contos. Dou software. António Fernandes, R. Poeta Milton 6, 3.º Esq.—1100 LISBOA

TIMEX 2048 + JOYSTICK QUICK SHOT II Turbo + Jogos, tudo como novo, e só 20 000 escudos. Contactar: João Marcos. Telef. (01) 7599327

SPECTRUM + 2 128K com JOYSTICK SINCLAIR e 80 jogos por 32 000$00.
Telefone: 2512093

Estudos de Psicologia Esotérica e Personalidade (nome e data/nascimento) só 500$00. Envie para GEOCEU / Apartado. 4278—4004 PORTO Codex

Todo o tipo de Software para PC. Sérgio Cónim, telef. 805880—1900 LISBOA

Jogos e utilitários para 2X SPECTRUM e PC IBM Compativel. Contacte: INFORAMS. Av. Elisyo de Moura, 397-2.º.
Telefone: 039-716949

Vendo todo o tipo de Software em cassete e diskette para mais informações contacte: Carlos Nuno Alves de Oliveira, Bairro de baixo, Ferreiros
4700 BRAGA

Programas e jogos para PC'S. Pedir lista e enviar selo para resposta para: Urb. Portel a lot.41 5.º Esq.—2685 SACAVÉM

Faço programas em DBASE BT, Pascal, Basic. Por encomenda, barato. Jorge Augusto. Pta. Rainha D. Amélia, 172 r/c
4400 V. N. GAIA

Monitor Mono por Policromático. Dou mais 20 000$00. (AMSTRAD) Pedro Estácio. R. D. João Castro, 185
RIO TINTO—Telef. 9894314

Drive para PC Compatível, 5 1/4 por 10 000$00. Pedidos a INFOR-SISTEM, apartado 125/89
2670 LOURES

Programas para PC. Mais de 12 programas, desde 700$00. Contactar: Jorge Augusto. Pta. Rainha D. Amélia, 172 r/c—4400 V. N. GAIA

Programas e jogos para PC. Pedir lista para: Paulo Jorge F., Largo da Republica n.º 6
8300 SILVES

Interface 1+MIC no Drive+ Waffer. Tudo por apenas 4 500$00.Contactar: Jorge Augusto. Pta. Rainha D. Amélia, 172 r/c—4400 V. N. GAIA

Programa de seguros para mediadores. (Apolices, recibos, contas, etc...)—Contactar: Luís Madaleno
Telefone: (01)2592260 (19h-21h)

Programas desde 500$00, mais de 10 programas, 100$00 para catálogo para: Jorge, Pta. Rainha D. Amélia, 172
4400 V. N. GAIA

AMSTRAD CPC 464+Disk Drive, Software, livros. Rogério Gonçalves, R. D. José Carcomo Lobo, 17 2.º Esq. Lavradio—2830 BARREIRO
Telef. 2043336

AMSTRAD 1640 Mono HD+ Windows, DBASE4. Outros—250 contos, Pedro Estácio, R. D. João Castro, 185 4435 Rio Tinto—Tel. 9894314

SPECTRUM +2+Monitor Monocromático + JOYSTICK + 5 jogos. Cinquenta mil escudos. Travessa Cabo Luís, 10—Esgueira—3800 AVEIRO
Telef. 312441

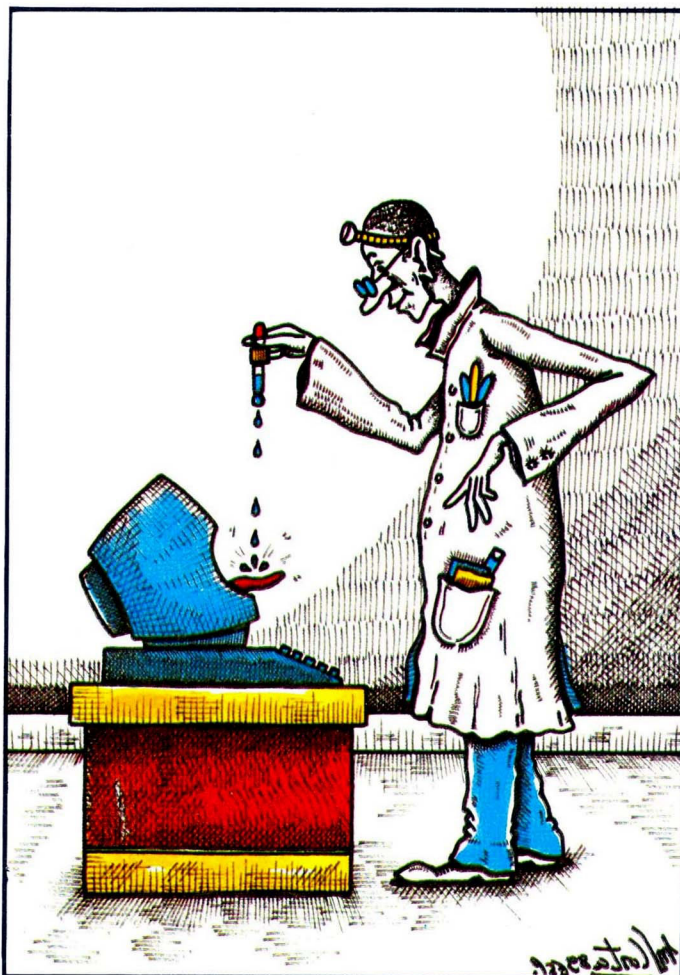
AMSTRAD PC1512 DD-MM—Telefone—(056) 61492
OLIVEIRA DE AZEMEIS

ATARI 1040ST+Rato+Ecrã SMI24 Mono+Software no valor de 170c. Tudo por 170c. Paulo Silva. Cam. Zamboeiras 1/2 Canidelo—4400 V. N. GAIA

PC1512, CGA Mono+Disco Rigido 30Mb+Impressora A4+ Software+Manuais+Disks. Preço 265cts. Av. D. Nuno Alv. Pereira, 166-4.ºDt.º
2800 ALMADA

Software PC em disq. 3,5 ''DD e HD mais de 300 títulos. José Orlando Pereira. Av. Dr. Trigo Negreiros, 237
5370 MIRANDELA

Optima oportunidade—AMSTRAD 1640 DD como novo pela melhor oferta. Ultimas novidades em Software grátis. Tel. (063) 32074 (20:24h) AMSTRAD

Todo o tipo de Software, enviar lista ou morada para: Nuno Pinheiro, Crestelo—6270 SEIA

COMPRO/TROCO/VENDO

Software para PC, cada diskette gravada 300$00. Tenho o Roger Rabbit. Carlos Jorge. Rua Olimpio Cabral n.º 94
5370 MIRANDELA

Todo o tipo de Software para PC. Bom e barato!!! Contacte: INFORAMS, Av. Elisyo de Moura, 397-2.º Dt.º
Telefone: (039) 716949

Programas e/ou jogos. Pascal Basic e DBbase são algumas das linguagens. Ricardo Pinto, Travessa Fernandes Tomás
OVAR—Telefone: 53408

Programas todo o tipo. Lista com mais 200 títulos. Envia a tua. Contacte: Guilherme Teixeira. R. Luciano Cordeiro, 14
5370 MIRANDELA

FDD 300 por 2 contos. Rua da Tranquilidade, lote 4A Charneca da Caparica 2825 CAPARICA
Software PC (utilidades e jogos) a partir de 300$00. Contactar: Paulo.Pr. Simão Veiga, T1-CB 8ESt.º António de Cavaleiros —2670 LOURES

TC 2068. Preço—20 contos. Ofereço programa contas bancárias+jogos. Contactar: Luís Morais. Telefone: (053) 25983 (fora das horas de serviço)

SPECTRUM +3 128K mais diskettes mais cassetes mais multiface 3 mais Joystick (bom estado), tudo por 35 000$00—Telefone: 765274

Impressora LQ500 AMSTRAD, EPSON 1050. Ambas novas, bom preço. Facilito o pagamento. Tel. (02) 308628. Trata com Duarte

AMSTRAD 164 + ecran polic. EGA/CGA/HERCULES + D 20MB + Software (+ 100c.). Melhor oferta.
Telef. (02) 812677. Jorge Prata

TIMEX 2068 + manual em Português + 150 jogos. Contactar: Tomé Pessegueiro—dias úteis. Tel. 26719 (044)—LEIRIA

Executo programas por encomenda, Pascal, Cobol, DBase, Basic, telef. (043) 63125. Joaquim J. Rolha—Valverde—2100 CORUCHE

Todo o tipo de Software para PC. Contacte: Sérgio Henriques. Tel. 7594200—1700 LISBOA

PC—Soft Clube. Vendemos todo o tipo de Software e (barato) para PC. Contactar o telefone 759420 ou 7581677 da rede de Lisboa. (1700 CP)

Executo trabalhos dactilografia e computador. Contactar tel. 29629 Coimbra. António Luís Jogos para o 1512 em diskettes de 5 1/4''.

Enviar 100$00 para lista. Computer Dream, Apartado 19 Valbom—4421 GONDOMAR CODEX

APPLE //E. Monitor Fosf. verde, dois drives com muito Software. Contactar: J. Polonia em Viseu—Telefone: (032) 20669 (depois das 20h)

ATARI 130XE (128K)
Disk Drive 51/4
Monitor Zenith.
Muito Software
Preço 90 000$00—Telefone: 9120035

2X SPECTRUM +48K + interface + Joystick Quicksnot2 + gravador + 5 jogos por 12 500$00. Contactar: Luís Eduardo—tel. 9329568

2X SPECTRUM 48K + Joystick + gravador. Contactar: Ricardo Frada—Tel. 9317306. 10 000$00

ZX SPECTRUM 48K em bom estado, cassetes como novas. Contactar: Tel. 565487 PORTO

ZX +3 c/ Software 45c, ZX 48K +204 + Light Pen + 15 cassetes + acessórios 25c. Ambos 60c. c/ tudo. Telefone: 2114441

YAMAHA DX11 + FENDER BULLET ''novo''. Preço imbatível possível troca contactar José Manuel S. Ferreira (032) 63591—3530 MANGUALDE

Software para SPECTRUM +3. Vendo também jogos para o SPECTRUM 48K. Escrever para: D.M.C. Apartado 28
4560 PENAFIEL

Interface 1 + Microdrive + 9 Cartrid. incluindo Tasword 3+ Software utilitário—15 000$00 —Telef. (049) 315912 depois das 19 horas

+3 com processador texto, folha cálculo, zip 2AP, Dlan, Tas-sign CP/M Plus, Base dados, Music Box. D.M.C. Apartado 28
4560 PENAFIEL

PC200: COMPATÍVEL? SIM.

—Ao folhear com mais calma a revista, venho deparar na publicação de Setembro do ano transacto, algo que me despertou o interesse. Muito propriamente na página 12, no que diz respeito à evolução dos produtos AMSTRAD, é noticiado para breve o lançamento de um computador doméstico 16 bits, compatível com IBM PC, conectado aos aparelhos de TV e trabalhando com disquetes de 3''1/2.

É sobre este computador que eu desejava obter elementos, acerca do seu preço, condições de pagamento, possibilidades de expansão, programas que pode utilizar, etc.

COVILHÃ

AM:—O computador em questão é o AMSTRAD SINCLAIR PC200. Trata-se de um compatível IBM, com uma ou duas unidades de disquetes de 3''1/2 (720K) e podendo também instalar uma hard-card.Neste computador poderá utilizar todo o software disponível para a família dos IBM-compatíveis.

WINDOWS NO PC1640

—Tenho um AMSTRAD 1640 ECD. Com ele tenho trabalhado muito bem, utilizando as WINDOWS V1.03, e o drivers para o MOUSE da AMSTRAD de DAVID BACK. Agora tenho as WINDOWS V2.03, mas não consigo instalar nem o mouse nem o próprio programa WINDOWS. Desejava que me ajudassem a resolver este problema.

Acácio Lobo

AM:—Ao configurar o WINDOWS escolha ecrã EGA com mais 64KB e o BUILT-IN MOUSE.

A COMPATIBILIDADE DAS ''HARD CARDS''

—A propósito das ''hard cards'', há alguma garantia da sua adequabilidade, por exemplo, ao ''PC1640 2D''? Poderá haver dificuldades/incompatibilidades a ter em conta na aquisição? Para quando a sua venda no ''Clube''? Concretamente, a placa anunciada (Alfa Sistemas Lda.) na capa interior da revista é compatível?

Angelo A. Vaz

AM:—Qualquer ''hard-card'' desde que seja para funcionar num PC compatível é perfeitamente adequada. Existe a vantagem de que podemos utilizar 2 drives (qualquer combinação 5''1/4, 3''1/2) e um disco ''hard-card'', beneficiando de um drive extra que não poderia existir se se optasse pela instalação de um disco rígido interno.
De momento, não está prevista a sua venda no Club AM, pelo que sugerimos um contacto com a Alfasistemas ou outra empresa do género.

DRIVE DE 5,25'' NÃO SERVE PARA O PS/30

—Gostava de saber se o produto (drives 5''1/4) ref.ª 903 se pode instalar num IBM PS/30 (os bons e maus aspectos).

Miguel Pedro F. Salvessa Rato

AM:—O Drive de 5''1/4 anunciado na revista A.M. não pode ser instalado num IBM PS, modelo 30, porque é um drive interno. Para que o PS 30 possa ter um drive de 5''1/4, é necessário adquirir um para montagem externa.

COLECÇÃO GEM SÓ PARA PC 1512 E 1640

—A colecção ''GEM'' apresentada no Clube AM da revista n.º 22 funciona sem problemas no AMSTRAD PC 2086?

Carlos Duarte

AM:—A colecção ''GEM'' do PC1512 não funciona na série PC2000, no entanto poderá utilizar uma versão para o PC1640.

O COMPUTADOR EM APLICAÇÕES SÉRIAS

FRUSTRAÇÃO

Há uma pequena história que se ouve nos meandros dos avançados laboratórios electrónicos e informáticos; conta-se, em termos francamente utópicos e futuristas que, uma vez criado o primeiro Computador dotado de inteligência artificial, todos aqueles que levaram avante a sua arrojada concepção, se prostraram perante a máquina, com a finalidade de fazer o grande e derradeiro teste. Os olhares convergiram para o Eng. responsável pelo projecto, e este sentiu-se incumbido, pelos seus colegas, a ligar, mais uma vez, a sua reconhecida capacidade de administrar as células cerebrais. Pouco depois, de entre os seus lábios, surgia a pergunta: ''Deus existe?...''

Passado algum tempo de profundo e enervante silêncio, a máquina ligou os seus circuitos e respondeu: ''Agora... existe!''

É que esta história, plena de fantasia, só pretende elucidar o leitor sobre o propósito deste artigo... levá-lo a pensar...! Uma máquina assim não passa de mera e pura ficção. Há muito que o Homem procura conceber o Computador Inter-activo, que mantenha um diálogo lógico com o utilizador... mas, até hoje, ainda não se passou das pesquisas. Em países como os E.U. ou o Japão já se preparam os circuitos que farão parte do chamado Computador da 5.ª geração, onde, entre outras características, se pretende: semi-consciência, visão, tacto..., quase que uma lista completa de sentidos e ''sentimentos''. Tudo aquilo que o aproxime, o mais possível, dos descendentes de Adão. Mas nunca um Computador que pense ser Deus ou Napoleão!...

Contudo, temos de aceitar que se o Computador de Hoje tivesse sentimentos decerto se sentiria frustrado quando as suas capacidades são, regra geral, canalizadas para programas tão fúteis quanto jogos. Mas... acalmem-se os entusiastas dos ''VIDEO-GAMES'', não sou, nem pretendo ser, um excessivo fanático por ocupações sérias e profissionais. Longe disso! Apenas penso que uma máquina tão potente, quanto um Computador, não deveria ter a sua programateca limitada, infantilmente, a jogos-de-vídeo.

É evidente que isto, apesar de geral. não constitui uma regra, e a muitos jogos são reconhecidas bastantes qualidades, indo o mérito principal para aqueles que incrementam substanciais melhoramentos no campo dos reflexos, perícia e raciocínio Humano. Tudo isto resulta num forte ''push'' no processo de evolução dum indivíduo, o que, diga-se de passagem, é o, ou um dos objectivos perseguidos...: EVOLUIR! Quanto a isto: ponto final. Parágrafo.

Só que tal não pode, nem deve, determinar uma limitação. Invariavelmente, todo o jogo se torna chato e, no fim, lá se vão mais umas quantas notas do Banco de Portugal, para adquirir a última (? pseudo) novidade, uma vez que aquele alucinante joguinho, que nos ocupava os fins-de-tarde, acabou de passar à História. Mas, a história é sempre a mesma!...

Porém, bem lá no fundo, todos ficamos gratos a Sir Clive Sinclair, um dos ''pais'' da microinformática, por colocar os Computadores ao alcance dos nosso cérebros, como igualmente das nossas bolsas. Agora pergunto: seriam exactamente jogos em que pensava o tio Clive (como, carinhosamente, o tratam na Grã-Bretanha) quando projectou, junto com outros engenheiros e técnicos da Sinclair Research, a série ZX, a partir do microprocessador (CPU) Z80? Aí já coloco as minhas sérias dúvidas.

Não faltou muito para que, em '82, as pesquisas da Sinclair resultassem no famoso ZX SPECTRUM. Se juntarmos as cores, a resolução gráfica e o som... acabaremos por encontrar um ''grande'' Computador, ao preço duma calculadora científica, do tamanho dum livro de bolso.

Contudo, haviam certos limites...; a memória (no início de 16K, depois de 48), a baixa, e muito indigna, velocidade de acesso a programas em cassete, e o teclado, tornavam este Computador um fruto proibido, para quem pensava dar-lhe um uso mais profissional.

É aqui que entra o AMSTRAD!...

Depois de adquirir a Sinclair, a Amstrad ousou—como é seu apanágio—promover o SPECTRUM a uma outra categoria. Em vez daquele metálico,e mal escolhido, teclado de borracha, surgiu com um muito mais profissional. Um gravador (como no CPC 464) ou uma unidade de disquete (como no CPC 664) começaram a fazer parte compacta do mesmo aparelho, como sendo mais duas das muito entusiasmantes opções adicionais pela AMSTRAD.

Com tudo isto surgem-nos algumas questões: Porquê mais memória? Porquê a unidade ''Disquet Drive'' (FDD)? Porquê um melhor teclado?... Em suma: Porquê um ''look'' tão profissional numa máquina supostamente, de jogos?... Porquê, concerteza, a AMSTRAD reconheceu nas características do SPECTRUM um forte candidato a aplicações sérias. Estalamos os dedos. Encontramos a resposta. A chave é: PROFISSIONALISMO! Tão simples quanto isso.

É verdade que passaram algumas versões até surgir o +3. Todavia valeu a pena esperar... Este tempo deixou-nos uma herança; um versátil microcomputador para toda a família: o ZX SPECTRUM 128K +3, com o cunho... AMSTRAD!

José A. de Matos

Impressoras Star

LC 10, LC 10 cor e LC 24-10
o melhor...
também para si!

LC 10

9 Agulhas

Velocidade em modo Draft de	120 cps Pica
	144 cps Elite
NLQ	30 cps Pica
	36 cps Elite

4 Fontes de caracteres Standard em NLQ, seleccionáveis por painel,

Itálico
Tractor anterior
Alimentação semi-automática A4
Função paper park
Emulação ESCAPE e IBM
Proprinter II
Opção alimentador automático A4 (ASF)

LC 10 Cor

24 Agulhas

Velocidade em modo Draft de	120 cps Pica
	144 cps Elite
NLQ	30 cps Pica
	36 cps Elite

4 Fontes de caracteres Standard em NLQ, seleccionáveis por painel,

Itálico
Tractor anterior
Alimentação semi-automática A4
Função paper park
Emulação ESCAPE e IBM
Proprinter II

Cassette de 4 cores, preto, vermelho, azul amarelo p / impressoras de 7 cores diferentes: preto, vermelho, azul, violeta, amarelo, laranja e verde

Opção alimentador automático

LC 2410

24 Agulhas

Velocidade em modo Draft de	142 cps Pica
	170 cps Elite
LQ	47 cps Pica
	57 cps Elite

4 Fontes de caracteres Standard em NLQ, seleccionáveis por painel,

Itálico
Tractor Anterior
Alimentação semi-automática A4
Função paper park
Emulação ESCAPE e IBM
Proprinter X 24 (NEC P6)

Opção alimentador automático A4 e RAM Cartridge 32KB

Distribuidor Oficial

Rua Ana de Castro Osório,
2-B/4-B (Quinta da Luz)
1500 LISBOA Portugal
Apartado 4513
1511 LISBOA CODEX
Tels. 715 12 37
714 01 28 - 714 46 46
Telex 65027 RIGAMA P
Telefax 714 42 64

a sua Impressora

CONSULTE O SEU REVENDEDOR